APPENDIX B

A SYSTEM FOR OBJECT DETECTION AND MOTION CLASSIFICATION IN COMPRESSED AND UNCOMPRESSED DOMAINS

Inventor(s) Name(s)*	Princeton Univ. Address	Princeton Telephone	Home Address
Wayne H. Wolf	Department of Electrical Engineering, E-Quad Building, Room B226	(609) 258 1424	Princeton, NJ
Ibrahim Burak Ozer	Department of Electrical Engineering, E-Quad Building, Room B326	(609) 258 7489	12-04 Deer Creek Drive, Plainsboro, NJ, 08536

General Description:

The invention is a new method which focuses on human detection due to its important applications in computer vision. The proposed retrieval method provides human detection and activity recognition at different resolution levels and connects low level features to high level semantics by developing relational object and activity presentations.

Utility:

The invention is used in a smart camera system that is under development at Princeton University. This smart camera is designed for use in a smart room in which the camera detects the presence of a person in its visual field and determines when various gestures are made by the person. As a first step toward a VLSI implementation, Trimedia processors hosted by a PC, are used. Other possible applications of the proposed method are content based image retrieval and similarity ranking in digital image/video libraries, surveillance systems and vehicle tracking.

Novelty:

The invention addresses the problem of object detection and activity recognition in compressed domain in order to reduce computational complexity and storage requirements. A new algorithm for object detection and activity recognition in JPEG images and MPEG videos is developed and we show that significant information can be obtained from the compressed domain in order to connect to high level semantics. This approach differentiates from previous compressed domain object detection techniques where the compression algorithms are governed by characteristics of object of interest to be retrieved. An algorithm is developed using the principal component analysis of MPEG motion vectors to detect the human activities; namely, walking, running, and kicking. Object detection in JPEG compressed still images and MPEG I frames is achieved by using DC-DCT coefficients of the luminance and chrominance values. The performance is dependent on the resolution especially for human detection where skin region extraction is crucial. For lower resolution and monochrome images it is demonstrated that the structural information of human silhouettes can be captured from AC-DCT coefficients. To increase the accuracy and to obtain more detailed information, the extraction of low level features from images and videos using intensity, color and motion of pixels and regions is done in uncompressed domain. Local consistency based on these features and geometrical characteristics of the regions is used to group object parts. The problem of managing the segmentation process is solved by a new approach that uses object based knowledge in order to group the regions according to a global consistency. A new model-based segmentation algorithm is introduced that uses a feedback from relational representation of the object. Object detection is achieved by matching the relational graphs of objects with the reference model. The algorithm maps the attributes, interprets the match and checks the conditional rules in order to index the parts correctly. The major advantages can be summarized as improving the object extraction by reducing the dependence on the low level segmentation process and combining the boundary and region properties. Furthermore, the features used for segmentation are also attributes for object detection in relational graph representation. This property enables to adapt the segmentation thresholds by a model-based training system. The major contribution of the overall algorithm is to connect available data in compressed and uncompressed domain to high level semantics. The proposed hierarchical scheme enables working at different levels, from low complexity to low false rates. This work provides a first and critical step towards the direction of real-time implementation of a human detection and activity recognition system. The adaptation of the proposed model for a smart camera system with multiple cameras is part of our current research.

A number of groups have developed smart rooms and buildings that track people. Early approaches used beacons carried by the subjects. However, a system that uses video avoids the need for beacons and allows the system to recognize gestures that can be used to command the operation of the smart room. A video-enabled smart room uses multiple smart cameras that both capture different views of the area and analyze the activity in their field of view. Tracking people and identifying what they are doing walking, making gestures, etc. is a challenging problem that requires the application of a number of different algorithms. The majority of research in human identification concentrates on algorithm development and is done in non-real time. Developing real-time human activity recognition systems requires simultaneous study of algorithms and architectures. Architectural information generally places bounds on the amount of processing power available and may suggest that some types of algorithms are more efficient than others. Knowledge of the structure of the algorithms and data is essential to making the most of the available architectural resources. The long-term goal is an integrated smart camera that includes a sensor, on-board processing, and on-board memory. Heterogeneous multiprocessors are the most suitable architectures for smart cameras. In order to gain experience with possible architectures, a heterogeneous multiprocessor using a PC as a host and VLIW processors for video operations is constructed. This platform allows us to evaluate algorithms running on real-time data and to make measurements that would be too expensive to conduct using simulation.

Method:

A) Process: The first and second parts of the process are object and activity detection requiring minimal decoding of compressed data in the proposed hierarchical method. Most object detection and human activity recognition techniques are done in the uncompressed domain and depend on proper segmentation of the body. The major contribution of the overall algorithm is to connect available data in compressed domain to high level semantics. The first part of the compressed domain process covers the principal component analysis of MPEG motion vectors to detect the human activities; namely, walking, running, and kicking. The second part corresponds to object detection in JPEG compressed still images and MPEG I frames. The algorithm uses DCT coefficients of the luminance and chrominance values obtained from the compression algorithms. The third part covers the human detection and posture recognition in uncompressed domain where a graph matching method with a model-based segmentation is proposed in order to connect low-level features to high level semantics by reducing the dependency on the feature extraction. First step in the uncompressed domain processing is the background elimination and color transformation. This step is the transformation of pixels (m by n) into another color space regarding to the application. For example transforming the RGB values into YUV components takes 5 additions and 8 multiplication for each pixel. Background elimination is performed by using these transformed pixel values. Our assumption for the background elimination is that the background is known and there is no change in the lighting conditions during the whole test sequence. Second step is the skin area detection. Skin areas are detected by comparing color values to a human skin model. We use Farnsworth nonlinear transformation in order to obtain uniform circular color differences. However, prior knowledge about the camera system and background increases the robustness of simpler skin color models suitable for real-time applications. We use YUV color model where chrominance values are down-sampled by two. Next step is the segmentation of non-skin areas and connected component algorithm. An object usually contains several sub-objects that can be obtained by segmenting the object of interest hierarchically into its smaller unique parts. The foreground regions that are adjacent to detected skin areas are extracted and corresponding connected components are found. We combine the meaningful, adjacent segments and use them as the input of the following algorithm steps. Contour following is the fourth step of the algorithm in the uncompressed domain. We apply the contour following algorithm that uses the 3x3 filter to follow the edge of the component where the filter can move in any of 8 directions to follow the edge . Each contour of size c_i is then fitted to a superellipse with 5 parameters by a Levenberg-Marquardt minimization method. Even when human body is not occluded by another object, due to the possible positions of non-rigid parts a body part can be occluded in different ways. For example, hand can occlude some part of torso or legs. In this case, 2D approximation of parts by fitting superellipses with shape preserving deformations provides more satisfactory results. It also helps to discard the deformations due to the clothing. Global approximation methods give more satisfactory results for human detection purposes. Hence, instead of region pixels, parametric surface approximations are used to compute shape descriptors. Graph matching is the last step of the uncompressed domain algorithm. Each extracted region modeled with ellipses correspond to a node in the graphical representation of human body. Face detection allows to start initial branches efficiently and reduces the complexity. Each body part and meaningful combinations represent a class (ω) where the combination of binary and unary features are represented by a feature vector (X) and computed off-line. Note that feature vector elements of a frame node computed online by using superellipse parameters, change according to body part and the nodes of the branch under consideration. For example, for the first node of the branch, feature vector consists of unary attributes. The feature vector of the following nodes includes also binary features dependent on the previous matched nodes in the branch. For the purpose of determining the class of these feature vectors a piecewise quadratic Bayesian classifier with discriminant function g(X) is used. The generality of the reference model attributes allows the detection of different postures while the conditional rule generation (r) decreases the rate of false alarms. The computations needed for each node matching are then a function of the feature size and the previously matched nodes of the branch under consideration. The marked regions are tracked by using superellipse parameters for the consecutive frames and graph matching algorithm is applied for new objects appearing in the other regions. This methodology was originally developed using Matlab; we are now adapting it for real-time multiprocessing.

B) Assembly: For real-time application, we use two Trimedia processors on two PCI cards attached to a host PC. Each Trimedia evaluation board includes a TM32 processor, local memory, and analog video input and output. Most video operations are performed on the on-board memory. The TM32 can also talk to the host PC using PCI transfers. The TM32 is programmed using the Trimedia C compiler running on the host PC. The Trimedia evaluation board is designed to support multiprocessing. TM32s can communicate via shared memory using the on-board memories without communicating directly with the host.

Limitations:

The detection of the objects becomes a more difficult problem for complex scenes with busy background or many objects with occlusions and shading. Occlusion and shading problems can be solved by using multiple cameras and camera setup can help to eliminate the false detection due to the busy background. Our assumption for the background elimination is that the background is known and there is no change in the lighting conditions during the whole test sequence.

Experimental Verification:

To evaluate the system performance for the activity recognition in compressed domain, several sequences with different activities are used. The results show that MPEG motion vectors corresponding to three human body sub-regions can be used for detection and recognition of human activity. Each test sequence gives the minimum normalized distance with its corresponding training set. The performance of the algorithm depends on the temporal duration of the observed activity. The detection rate for 141 non-occluded pedestrian images in frontal or near-frontal views for low resolution and monochrome JPEG images is 82%. In order to train our system, we use 800 positive examples and 600 negative examples with a bootstrapping algorithm. We achieve a correct detection rate of approximately 80%. Our approach has the advantage of using the available data in standard compression algorithms and gives highly accurate detection results. The performance of the proposed algorithm for non-rigid objects is given for 42 test images with human bodies for front and side views which are chosen from different sources. Since bending deformation increases the computational complexity, its value is set to zero and the computations are done using the tapering deformation. In the model file, the adjacency information between parts is given as; head-torso, upper arm-torso, leg-foot, lower arm-hand, etc. Under the assumption that feature vectors have Gaussian distribution, their mean and variance are determined during supervised learning. The overall algorithm performance is obtained by computing the correct, false, and miss detection of the body parts in the test images. The preliminary results show that 70.27% of the body parts are correctly and 18.92% are falsely classified. The remaining 10.8% is the miss detection. In order to determine the posture of the persons in the still images and video sequences, the binary features of the corresponding matched node pairs are used after the classification. For example, the angle between the image node matched to torso and image node matched to arm informs how much arms are open.

Our algorithmic pipeline clearly performs a wide range of disparate operations: 1) pixel-by-pixel operations, such as color segmentation; 2) pixel-region operations, such as region identification; 3) mixed operations, such as superelllipse fitting; 4) non-pixel operations, such as graph matching. We start with operations that are clearly signal-oriented and move steadily away from the signal representation until the data is very far removed from a traditional signal representation.

Abstract:

We propose a hierarchical retrieval system where shape, color and motion characteristics of human body are captured in compressed and uncompressed domains. The proposed retrieval method provides human detection and activity recognition at different resolution levels from low complexity to low false rates and connects low level features to high level semantics by developing relational object and activity presentations. The available information of video compression algorithms are used in order to reduce the amount of time and storage needed for the information retrieval. The principal component analysis is used for activity recognition using MPEG motion vectors and results are presented for walking, kicking and running to demonstrate that the classification among activities is clearly visible. For lower resolution and monochrome images it is demonstrated that the structural information of human silhouettes can be captured from AC-DCT coefficients. The system performance is tested on 40 images that contain a total of 126 non-occluded frontal poses and the algorithm can detect 101 of them correctly. The finest details in the images and video sequences are obtained from the uncompressed domain via model based segmentation and graph matching for an in depth analysis of human bodies. The detection rate for human body parts is 70.27% for images and sequences including human body regions at different resolutions and with different postures.

The adaptation of the proposed model for a camera system in real-time with multiple smart-cameras is part of our current research. We propose a prototype system for real-time human activity recognition and present our current view of the architectures that will be required for smart cameras. This smart camera is designed for use in a smart room in which the camera detects the presence of a person in its visual field and determines when various gestures are made by the person. As a first step toward a VLSI implementation, we use Trimedia processors hosted by a PC.

Possible Means of Commercialization:

Our long-term goal is a system-on-a-chip; an integrated smart camera that includes a sensor, on-board processing, and on-board memory. For example, MediaWorks and Clever Systems are two of the companies that design system-on-a-chips which will lower the costs, meet the performance and power requirements for multimedia applications. In order to make the embedded software, that is used by the chip, work at the required performance, power, and cost, system-on-a-chips are designed as custom multiprocessors. Multiple processors are the best way to ensure that real-time constraints are met in a time-efficient manner. Unlike the multiprocessors used for scientific computing, these system-on-a-chips are heterogeneous and application-specific. The types and number of computing units, memory, and interconnect are determined by the needs of the application. Only by tuning the architecture to the application can you get a cost- and power-effective solution. Standard CPUs and DSPs aren't system-on-a-chips because they don't solve a system problem. CPUs and DSPs are simply components in larger systems--they need memory, I/O, and multiprocessing to create full-fledged systems. Standard parts--CPUs, DSPs, FPGAs, ASICs--are useful for prototyping but don't solve the needs that system-on-a-chips address.

Attached Exhibits:

The attached exhibits each include disclosure directed to various aspects of the invention.

A Hierarchical Human Detection System in (Un)compressed Domains

I. Burak Ozer, *Member, IEEE*, and Wayne Wolf, *Fellow, IEEE*

Abstract

With the rapid growth of multimedia information in forms of digital image and video libraries, there is an increasing need for intelligent database management tools with an efficient information retrieval system. For this purpose, we propose a hierarchical retrieval system where shape, color and motion characteristics of human body are captured in compressed and uncompressed domains. The proposed retrieval method provides human detection and activity recognition at different resolution levels from low complexity to low false rates and connects low level features to high level semantics by developing relational object and activity presentations. The available information of standard video compression algorithms are used in order to reduce the amount of time and storage needed for the information retrieval. The principal component analysis is used for activity recognition using MPEG motion vectors and results are presented for walking, kicking and running to demonstrate that the classification among activities is clearly visible. For low resolution and monochrome images it is demonstrated that the structural information of human silhouettes can be captured from AC-DCT coefficients. The system performance is tested on 40 images that contain a total of 126 non-occluded frontal poses and the algorithm can detect 101 of them correctly. The finest details in the images and video sequences are obtained from the uncompressed domain via model based segmentation and graph matching for an in depth analysis of human bodies. The detection rate for human body parts is 70.27% for images and sequences including human body regions at different resolutions and with different postures.

Keywords

Image and Video Databases, Human Detection, Activity Recognition, Model-based Segmentation, Relational Graph Matching, Eigenspace Representation, MPEG, JPEG.

I. Introduction

THE rapid growth of multimedia information in forms of digital image and video libraries necessitates intelligent database management tools. Although the visual information is widely accessible, technology for extracting the useful information is still restricted. Traditional text-based query systems based on manual annotation process are impractical for today's large libraries requiring an efficient information retrieval system. The efficiency of such a system should be evaluated in terms of extraction of high-level semantics, information access time and allocation of bandwidth and storage. For this purpose, we propose a hierarchical retrieval system (Fig. 1) where shape, color and motion characteristics of objects of interest (OOI) are captured in compressed and uncompressed domains. This paper focuses on human detection due to its important applications in computer vision. The proposed retrieval method provides human detection and activity recognition at different resolution levels and connects low level features to high level semantics by developing relational object and activity presentations. The available information of standard video compression algorithms are used in order to reduce the amount of time and storage needed for the information retrieval. This differentiates our approach from previous work where the information retrieval applications for standard compression algorithms are restricted to index activity levels and track objects in videos while object and activity detection algorithms are implemented by using non-standard compression schemes governed by characteristics of objects. The finest details in the images and video sequences are obtained from the uncompressed domain via model based segmentation and graph

I. Burak Ozer and Wayne Wolf are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, U.S.A. E-mail: (iozer,wolf)@ee.princeton.edu.

matching for the analysis of human bodies. The proposed hierarchical scheme enables working at different levels, from low complexity to low false rates.

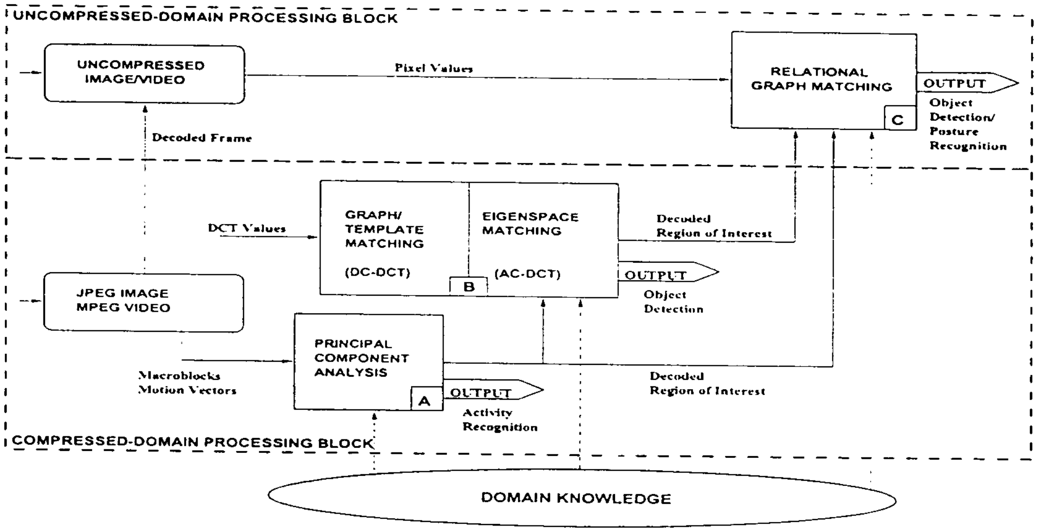

UNCOMPRESSED-DOMAIN PROCESSING BLOCK
UNCOMPRESSED IMAGE/VIDEO
Pixel Values
RELATIONAL GRAPH MATCHING
C
OUTPUT
Object Detection/ Posture Recognition
Decoded Frame
GRAPH/ TEMPLATE MATCHING
(DC-DCT)
EIGENSPACE MATCHING
(AC-DCT)
B
DCT Values
Decoded Region of Interest
OUTPUT
Object Detection
JPEG IMAGE MPEG VIDEO
PRINCIPAL COMPONENT ANALYSIS
A
Macroblocks Motion Vectors
OUTPUT
Activity Recognition
Decoded Region of Interest
COMPRESSED-DOMAIN PROCESSING BLOCK
DOMAIN KNOWLEDGE

Fig. 1. Overall algorithm.

An important issue in digital libraries is the query representation which is related to the user interface. Query by example is a method of query specification that allows a user to specify a query condition by giving image examples. Main features of an image can be given as shape, spatial relation, color and texture. Another method is to draw the shape of the object. Images are also retrieved by specifying colors and their spatial distribution in the image. User can also specify the movement of an object for video retrieval. If textual descriptions representing the content of images are available then a query by keyword can be performed. The proposed retrieval system is used to annotate video sequences and images that contain OOI (human) in order to enable text based queries and to retrieve detailed information about the OOI, i.e., activity/posture recognition.

Automatic annotation of images where an object of interest is present faces three major problems. One is the dependency of the object detection on the feature extraction process which is a complex task especially for cluttered scenes. The second is that the visual properties of images, that are described by feature vectors, are difficult to describe automatically with text. Therefore, the similarity retrieval connecting these vectors to high level semantics and using high level knowledge to improve feature extraction become an important issue. Finally, these processes should require a reasonable amount of computation time and storage.

Our retrieval system (Fig. 1) consists of two major blocks, namely uncompressed and compressed-domain processing blocks. In the compressed-domain processing block, we address the problem of object detection and activity recognition in compressed domain in order to reduce computational complexity. New algorithms for object detection and activity recognition are developed for JPEG images and MPEG videos to show that significant information can be obtained from the compressed domain in order to connect to high level semantics. Since our aim is to retrieve information from images and videos compressed using standard algorithms such as JPEG and MPEG, our approach differs from previous compressed domain object detection techniques where the compression algorithms are governed by characteristics of object of interest to be retrieved. An algorithm is developed using the principal component analysis of MPEG motion vectors to detect the human activities; namely, walking, running, and kicking [1]. Object detection in JPEG compressed still images and MPEG I frames is achieved by using DC-DCT coefficients of the luminance and chrominance values. The performance is dependent on the resolution, especially for human detection where skin region extraction is crucial. Therefore, for lower resolution and monochrome images we demonstrate that the structural information of human silhouettes can be captured from AC-DCT coefficients [2].

If the database consists of uncompressed images and videos then uncompressed-domain processing techniques are used. Furthermore, if a more detailed analysis of the retrieved information is needed, the region of interest extracted from a compressed image or video is further processed by using uncompressed-domain processing block. Therefore, the inputs to the Block C in Fig. 1 are the uncompressed database image or video sequence and decoded image or video frame of interest extracted by using compressed-domain techniques.

Our method extracts low level features from the regions extracted in the compressed domain or from uncompressed images and videos using intensity, color and motion of pixels [3]. Local consistency based on these features and geometrical characteristics of the regions is used to group object parts. We then take a new approach to the problem of managing the segmentation process by using object based knowledge in order to group the regions according to a global consistency and introduce a new model-based segmentation algorithm by using a feedback from relational representation of the object. The selected unary and binary attributes are further extended for application specific algorithms, namely an elaborate human skin color model and weak perspective invariants for articulated movements. Object detection is achieved by matching the relational graphs of objects with the reference model. The algorithm maps the attributes, interprets the match and checks the conditional rules in order to index the parts correctly. This method improves object extraction accuracy by reducing the dependency on the low level segmentation process and combining the boundary and region properties. Furthermore, the features used for segmentation are also attributes for object detection in relational graph representation. This property enables to adapt the segmentation thresholds by a model-based training system. The detailed algorithm of the graph matching process is given in Fig. 2.

Consider a recorded and MPEG compressed video sequence taken from a fixed camera surveying a passage. The first step will retrieve possible frames where people walk. This is achieved in the compressed-domain processing block (Fig. 1) by implementing the principal component analysis algorithm for MPEG motion vectors, obtained from 16x16 macro-blocks, that help to recognize walking activity (Block A in Fig. 1). If a walking person is detected to stop, second step will analyze the extracted region for posture recognition. This is achieved by using the information obtained

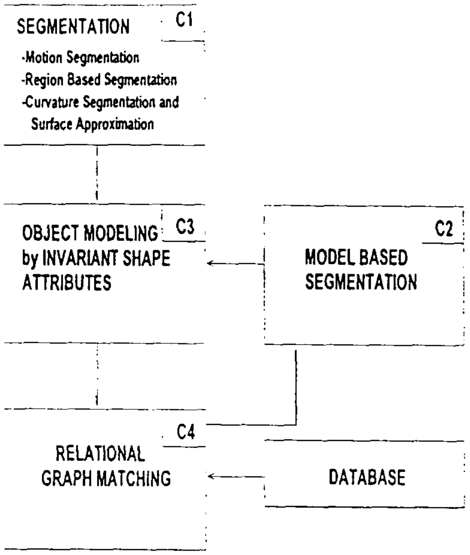

SEGMENTATION
C1
·Motion Segmentation
·Region Based Segmentation
·Curvature Segmentation and Surface Approximation
OBJECT MODELING by INVARIANT SHAPE ATTRIBUTES
C3
C2
MODEL BASED SEGMENTATION
C4
RELATIONAL GRAPH MATCHING
DATABASE

Fig. 2. Relational graph matching algorithm (Block C in Fig. 1).

from Block A and DC/AC-DCT coefficients for 8x8 blocks of the MPEG I-frames (Block B in Fig. 1). If a suspicious movement is detected, the third step will be a more detailed investigation of the region in the uncompressed domain. This is achieved by decoding the video frames of interest with suspicious movement and processing these frames further by using relational graph matching algorithm (Block C in Fig. 1).

Depending on the application, our proposed system can be used to a) retrieve different types of activities from the MPEG video database via the proposed method given in Block A, b) retrieve object of interest (human) from the compressed database images via the proposed method given in Block B, c) retrieve images of people with different postures from the database via the proposed method given in Block C.

This paper is organized as follows. Section II is a review of existing literature devoted to content-based retrieval systems in compressed and uncompressed domains. In Section III, we propose new algorithms for object detection and activity recognition in the compressed domain in order to reduce computational complexity and processing time. The first part of this section covers our new algorithm for principal component analysis of MPEG motion vectors to detect the human activities; namely, walking, running, and kicking. The second part corresponds to our proposed method for object detection in JPEG compressed still images and MPEG I frames. Possible human areas in the image are detected by using the DCT coefficients in a principal component analysis. Section IV covers the human detection and posture recognition in uncompressed domain. In this section we propose a new model-based segmentation algorithm and a new graph matching method in order to connect low-level features to high level semantics by reducing the dependency on the feature extraction. The experimental results are presented in Section V to evaluate the performance of each algorithm block in the uncompressed and compressed domains. Conclusions and suggestions for future research are offered in

Section VI.

II. Previous Work

This section reviews the information retrieval methods and systems for uncompressed and compressed domains.

A. Shape Retrieval

Many researchers have studied shape-based search. Shape based image retrieval is one of the hardest problems in general mainly due to the difficulty of segmenting objects of interest in images. The preprocessing algorithm determines the contour of an object depending on the application. Once the object is detected and located, its boundary can be found by using edge detection and boundary following algorithms [4]. The detection of the objects becomes a more difficult problem for complex scenes with busy background or many objects with occlusions and shading. If the object border is determined its shape can be characterized by its shape features. These feature vectors are generated by using a shape description method to characterize a shape. The required properties of a shape description scheme are invariance to translation, scale, rotation, luminance, and robustness to partial occlusion. Afterwards, shape matching is used in model-based object recognition where a set of known model objects is compared to an unknown object detected in the image using a similarity metric. Our description scheme is motivated by the well-known human perception theory and shape analysis techniques.

Shape similarity methods can be classified into two parts namely contour and region based techniques. Birchfield [5] states that every closed set in a plane can be decomposed into its two disjoint sets; the boundary and the interior according to elementary set theory. Since these two sets are mathematically complementary, the author claims that the failure modes of a tracking module focusing on the object's boundary will be orthogonal to those of a module focusing on the object's interior. Since the same concept can be applied to shape analysis, the combination of contour and region based shape descriptors are used in the proposed system.

A.1 Contour-based Techniques

A signature of a boundary may be generated by computing the distance from the centroid to the boundary as a function of angle [6]. Chang [7] constructs the distance function from the centroid to the feature points that are the points of high curvature. Another boundary representation technique is the curve approximation by utilizing polygonal and spline approximations. Bengston and Eklundh [8] proposes a hierarchical method where the shape boundary is represented by a polygonal approximation. Splines have been very popular for the interpolation of functions and the approximation of curves. They possess the beneficial property of minimizing curvature [9], [10].

Scale space techniques rely on the object representation at different scales. Witkin [11] proposes a scale space filtering approach which provides a useful representation for significant features of an object filtered by low-pass Gaussian filters of variable variance. Mokhtarian and Mackworth [12] uses the scale space approach as a hierarchical shape descriptor.

The major drawback of these techniques is the dependency on the extraction of the object boundary. Another problem is the difficulty to evaluate the similarity between the boundaries of objects with high within-class variance.

A.2 Region-based Techniques

The use of moments for shape description was proposed by Hu [13] who showed that moment based shape description is information preserving. An alternative transform approach is the Fourier transform of the shape. One of the disadvantages of these descriptors is that they do not reflect local shape changes. Leymarie and Levine [14] find the medial axis transform using snakes for active contour representation, high curvature points on the boundary, and symmetric axis transform. Superquadrics are widely used for modeling three dimensional objects in computer vision literature [15], [16]. Even when human body is not occluded by another object, due to the possible positions of non-rigid parts a body part can be occluded in different ways. Parametric modeling of image segments helps to overcome this problem and reduces the effect of the deformations due to the clothing.

As in the contour-based modeling, the performance of these techniques depend on the extraction of the object regions. Furthermore, higher order shape metrics is needed for the presentation of the complex objects. One solution is to decompose the object for its presentation as a combination of component shapes. The idea is to represent complex shapes in terms of simpler components. However, the shape decomposition should also create semantic segments for purposes of similarity retrieval of non-rigid objects.

B. Color Retrieval

There are two approaches for querying by color: by regional color and by global color [18]. Regional color corresponds to spatially localized colored regions within the scenes. Global color corresponds to the overall distribution of color within the entire scene. Color information can be represented as color sets that give selection of colors or color histograms that denote the relative amounts of colors. Different color space bases related to human color judgments can be used [70]: HSV color space by Smith [19] and Yu [20], LUV color space by Moghaddam [21], YES color space by Saber [22]. Color models play an important role in extraction of skin regions for human detection systems [5], [23]. However, color information alone is not enough for retrieval systems and should be used with shape and motion attributes for an intelligent retrieval system.

C. Motion Retrieval

Motion is mostly used to index videos according to their activity levels, to detect shot and scenes in compressed and uncompressed domains [24], [25], [26]. Human motion analysis is another main research area that uses motion for information retrieval [27], [28], [29], [30]. Most of the previous work in activity recognition are done in uncompressed domain after a proper segmentation of human body while motion information retrieval from compressed domain is restricted to index videos and track objects. Motion extraction in compressed domain and human activity recognition are reviewed in more detail in Section III.

Some of the information retrieval systems allow the user to make a query using motion as the key object attribute [33]. Motion is also used for several video content-based retrieval systems in compressed and uncompressed domain for specific applications e.g, sports video processing. Kurokawa et al. [34] retrieve scenes of soccer plays from several soccer video sequences. Motion is used to describe action of objects, interactions between objects and events using spatial and temporal relationships. Miyamori et al. [35] annotate tennis video where the court layout knowledge is used assuming that shots including tennis courts are preextracted. In Tan [36], the authors use camera motion to analyze and annotate basketball videos and browse for events such as wide-angle and close-up views, fast breaks, probable shots at the basket.

D. Retrieval Systems

Content based image/video indexing and retrieval has been researched by the governmental [38], [39] and industrial [40], [41] groups as well as at the universities [19], [20], [42], [43]. Different techniques are used based on image features such as shape, color, texture, motion or a combination of them. A survey of these retrieval systems can be found in Gupta [44] and Smeulders [45]. Some of these systems, described below, support query by keyword representing a semantic concept.

One of the systems is the Photobook [46] which is a software tool for performing queries on image databases based on image content and textual annotation. It basically compares features associated with images. Cypress-Chabot [47] integrates the use of stored text and other data types with content-based analysis of images to perform "concept queries". In Webseek [48], the images and video are analyzed using visual features (such as color histograms and color regions) and the associated text utilized to classify the images into subject classes. SEMCOG [49] system performs a semiautomatic object recognition and it aims at integrating semantics and cognition-based approaches to give users a greater flexibility to pose queries. One of the commercial systems is QBIC [40], which supports several basic image similarity measures such as average color, color histogram, color layout, shape and texture.

"Human" is one of the major objects of interest to be retrieved in the content-based retrieval systems. Great effort has been devoted to human recognition related topics such as face recognition in still images, and motion analysis of human body parts. Most of the previous work depend highly on the segmentation results and mostly motion is used as the cue for segmentation [28]. There has been very few work that are on the human recognition in still images and in compressed domain. Although Franke [50] and Papageorgiou [51] use a compact representation of the training sets that are suitable for cluttered scenes there is no direct correspondence between the low level features and body parts. Such a semantic representation is needed for high level applications and for occlusion problems. In another survey by Gavrila

[29], the segmentation problem is pointed out especially for detection of multiple and occluded humans in the scene.

Most of the previous work in human detection and activity recognition are done in uncompressed domain. Since image and video applications are generally represented in the compressed domain, such as JPEG or MPEG, there is a need for image/video manipulation and automatic content extraction in the compressed domain. As stated in Chang [52], for existing compression standards the compressed-domain image/video manipulation techniques can be used to help to solve the bandwidth and storage problem. Hence applications without expanding the coded visual content back to the large, uncompressed domain would reduce the need of large bandwidth and intensive computing. The use of available information in compressed video and images has been investigated mostly for video indexing, and shot and scene classification. In Yeung [24], hierarchical decomposition of a complex video is obtained using scaled DC coefficients in an intra coded DCT compressed video for browsing purposes. The technique combines visual and temporal information to capture the important relations within a scene and between scenes in a video. In Yeo [53], the authors examine the direct reconstruction of DC coefficients from motion compensated P-frames and B-frames of MPEG compressed video. In Dawood [25], an automatic scene classification scheme is proposed for MPEG videos. The scenes are divided into low, medium, and high texture and activity scenes.

MPEG motion vectors are used mostly to index videos (low-high activity) and track objects. The object detection in the compressed domain is more restricted since this application requires more detailed information. In Schonfeld [54], an object tracking algorithm is proposed using compressed video only with periodically decoding I-frames. The object to be tracked is initially detected by an accurate but computationally expensive object detector applied to decoded I-frames. Zhong et al. [55] automatically localize captions in JPEG compressed images and I frames of MPEG compressed videos. Intensity variation information encoded in the DCT domain is used to capture the directionality and periodicity of blocks. Wang [56] proposes an algorithm to detect human face regions from dequantized DCT coefficients of MPEG video. The algorithm uses the DC DCT values of chrominance, shape, and energy distributions of the face area. This method is suitable for color images with face regions greater than 48 by 48 pixels (3 by 3 MPEG macro-blocks). The authors extend their work in [57] in order to track and summarize faces from compressed video. The previous algorithm is used to detect faces and MPEG motion information is used with the Kalman filter prediction to track faces within each shot. The representative frames are then decoded for pixel domain analysis and browsing.

III. Human Detection and Activity Recognition in Compressed Domain

This section presents object and activity recognition in the compressed domain in order to reduce computational complexity and processing time (Fig. 1), compressed-domain processing block (Blocks A and B)). For large libraries, compressed domain image/video processing for existing compression standards can solve the problem of storage and intensive computing. In this work, new algorithms for object detection and activity recognition in JPEG images and

MPEG videos are developed. We show that significant information can be obtained from the compressed domain in order to connect to high level semantics.

The first (Fig. 1, Block A), and second (Fig. 1, Block B) parts of the proposed system are object and activity detection requiring minimal decoding of compressed data in the proposed hierarchical method. Most object detection and human activity recognition techniques are done in the uncompressed domain and depend on proper segmentation of the body. The major contribution of the overall algorithm is to connect available data in compressed domain to high level semantics.

The first part of this section covers the principal component analysis of MPEG motion vectors to detect the human activities; namely, walking, running, and kicking. The second part corresponds to object detection in JPEG compressed still images and MPEG I frames. The algorithm uses DCT coefficients of the luminance and chrominance values obtained from the compression algorithms.

A. Activity Recognition Using MPEG Motion Vectors

The activity recognition problem can be divided into two subparts: the first one is collecting satisfactory measurements and the second one is developing a recognition algorithm based on these measurements. Most of the related work use activity measurements from uncompressed images after a proper segmentation of human body parts. Our measurements are obtained from MPEG motion vectors for macro-blocks in intra-frames. Since the resolution of the motion vectors is one macro-block and there is no direct correspondence with the object parts and their motion, a robust and global model must be used. The corruption of data is another problem in MPEG motion vectors since some blocks can not be tracked during some frames. Overviews of research on human motion analysis can be found in Aggarwal [28] and Gavrila [29]. The major problems in the activity recognition is the scale, shift and projection changes between the model and the test data and segmentation dependency. One of the activity modeling methods proposed in Walter [30] is based on first order Markov model descriptions and continuous propagation of observation density distributions. Hidden Markov Models are used to predict the state transitions. In Rangarajan [31], speed and direction components of 2D trajectories are represented by scale-space images that are invariant Euclidean transforms. The outline of the human body is used to detect the periodical relative limb movement in Curio [32] by a template matching process. In these approaches, for each activity, a separate model is developed in order to compare with the observed activity. These approaches are robust to local transformations but lack a global detailed model to capture the variabilities.

Principal component analysis (PCA) method is one of the global approaches. PCA has been successfully used by Yacoob and Black [27] for human activity recognition in uncompressed video sequences. The authors use the motion measurements for segmented human body parts and recognize the articulated activities such as walking, kicking, and marching. They define these activities as atomic activities which satisfy two conditions. First one is that the movements are structurally similar for different performers, and second one is that the movements can be mapped onto a finite temporal window. For this reason, in this paper we study the detection of these activities. Our aim is to demonstrate that substantial information can be retrieved directly from compressed databases. Specifically, we extend PCA approach to recognize human activities such as walking, running and kicking in MPEG compressed videos.

In our method, first the moving regions are detected and then the motion vectors are grouped automatically by using the ratio of the human body parts. Hence the measurements do not correspond to the actual human body parts but to macro-block groups corresponding to human region. For the classification of moving regions, the neighboring blocks with a velocity greater than a predefined threshold are classified as one moving object. The following subsection covers the principal component analysis.

A.1 Principal Component Analysis of MPEG Motion Vectors

PCA is a dimensionality reducing technique used in pattern recognition. It reduces dimensionality by projecting the motion vectors to a new space spanned by the training data set. PCA was successfully used for face recognition. A compact representation of facial appearance is described in Kirby et al. [61], where face images are decomposed into weighted sums of basis images using a Karhunen-Loeve expansion. The eigenpicture representation has been used in Turk et al. [62] as eigenfaces for face recognition.

For training the system, several walking, running and kicking people sequences which are temporally aligned are used. For these sequences, the object region is extracted by grouping MPEG motion vectors. Then, the object is segmented to three parts (upper body, torso and lower body) according to the human body proportions. The mean of the motion vectors in horizontal and vertical direction is computed for the macro-blocks corresponding to each part (6 parameters) for a number of sequences T. A training set of k different examples for each activity forms matrix A of dimensions $6T \times k$. Then the singular value decomposition of the matrix A is computed to get the approximated projection of the exemplar vectors (columns of A) onto the subspace spanned by the $q < k$ basis vectors. Hence activity basis with parameters m are computed [27].

$$A = U\Sigma V^T \tag{1}$$

where A is the motion parameter matrix, U represents the principal component directions, Σ includes the singular values, and V^T expands A in principal component directions. To recognize the activities, an unknown sequence, other than test sequences of an activity which can be shifted and scaled in time is compared with the training set. The transformation function κ might model uniform temporal scaling and time shifting to align observations with exemplars. Let $D(t)$ be an observed activity, $[D]$ be the nT column vector, obtained first by concatenating the n feature values measured at t, and then concatenating $D(t)$ for all t. Let $[D]_j$ denote the j-th element of vector $[D]$. By projecting this vector on the activity basis, a coefficient vector ($\bar{c}$) is recovered, which approximates the activity as a linear combination of activity basis. For recovering the coefficients, the error has to be minimized:

$$E(\bar{c}) = \sum_{j=1}^{nT} \rho(([D]_j - \sum_{l=1}^{q} c_l U_{l,j}), \sigma) \tag{2}$$

where $\rho(x, \sigma)$ is an error norm over x, and σ is a scale parameter. Let $\kappa(\bar{a}, t)$ denote a transformation with a parameter vector $\bar{a}$ that can be applied to an observation $D(t)$ as $D(t + \kappa(\bar{a}, t))$. After Taylor series expansion of $D(t + \kappa(\bar{a}, t))$, the error function becomes:

$$E(\bar{c}, \bar{a}) = \sum_{j=1}^{nT} \rho([D_t(t)\kappa(\bar{a}, t) + D(t))]_j - \sum_{l=1}^{q} c_l U_{l,j}, \sigma) \tag{3}$$

Equation 3 is minimized with respect to $\bar{a}$ and $\bar{c}$ using a gradient descent scheme with a continuation method that gradually lowers σ. The normalized distance between the coefficients m_i from the training data set and coefficients of exemplar activities c_i is used to recognize the observed activity that is transformed by the temporal translation, scaling and speedup parameters [27]. The Euclidean distance is given as

$$d^2 = \sum_{1}^{q} (c_i/||\mathbf{c}|| - m_i/||\mathbf{m}||)^2 \tag{4}$$

where $\mathbf{c}$ is vector of expansion coefficients of an exemplar activity. The algorithm is applied for recognition of three activity classes: walking, running, and kicking. 10 training test sequences for each class are obtained from various sources for the side-view. The camera motion is assumed to be zero. In Fig. 3, some test frames from the activity training sets are displayed. The detection of the moving regions and the determination of the activities from the grouped MPEG motion vectors give a coarse information about the scene. Fig. 4 displays the motion vectors obtained from the intra-frames. Afterwards, these vectors are grouped by using the ratio of the human body parts to be used in PCA algorithm.

For a more detailed investigation, one may need additional information. DC-DCT coefficients and coefficient differences obtained from MPEG sequences in the compressed domain are processed in the next subsection.

Fig. 3. Frames from walking, running, and kicking man training sets.

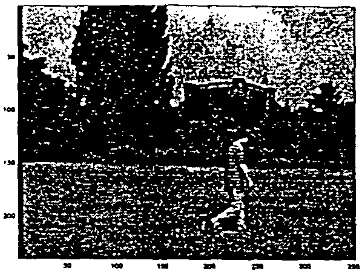


Fig. 4. Motion vectors for a walking man sequence.

A.2 Analysis of DC Differences

In this subsection, 8 by 8 block information (DC values) in the frames where human activity has been detected from the macro-block information (motion vectors), are used. The difference of the DC values for 8 by 8 blocks between consecutive frames are computed and the difference image is binarized by threcholding. To train our system, several human activity sequences from side-view with the similar camera distance, human motion direction and velocity are used. In order to find the template for each body position during one activity period, the mean of the moving regions, corresponding to these positions, are calculated. One of the templates is shown in Fig. 5. The classification is done by using a basic template matching measure. Note that the mirror image of the template is also used. For every DC-DCT difference frame, the blocks are compared to the activity templates (Fig. 6) For scale change invariance, the moving block regions with different scale parameters are scaled and the matching value for each scale factor is calculated.

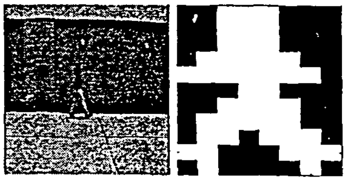


Fig. 5. Left: Walking position in the uncompressed image, Right: Template corresponding to this position.

B. Object Detection in JPEG Images and MPEG I Frames

Our proposed method operates on the I-frames of MPEG video or JPEG images, using DCT coefficients of image blocks. DCT compressed images encode a two-dimensional image using the DCT coefficients (c_{uv}) of an LxL image region ($I_{xy}, 0 \leq x < L, 0 \leq y < L$):

$$c_{uv} = K_u K_v \sum_{x=0}^{L-1} \sum_{y=0}^{L-1} I_{xy} cos \frac{\pi u(2x+1)}{2L} cos \frac{\pi v(2y+1)}{2L} \quad (5)$$

In Eq. 5, u and v denote the horizontal and vertical frequencies and $K_u = 1/\sqrt{L}$ if $u = 0$ and $K_u = \sqrt{2/L}$, otherwise. The AC components (c_{uv}, $u \neq 0$ or $v \neq 0$) capture the spatial frequency and directionality properties of the image block. From the regenerated array of quantized coefficients, the DCT coefficients are extracted. Although they are quantized,

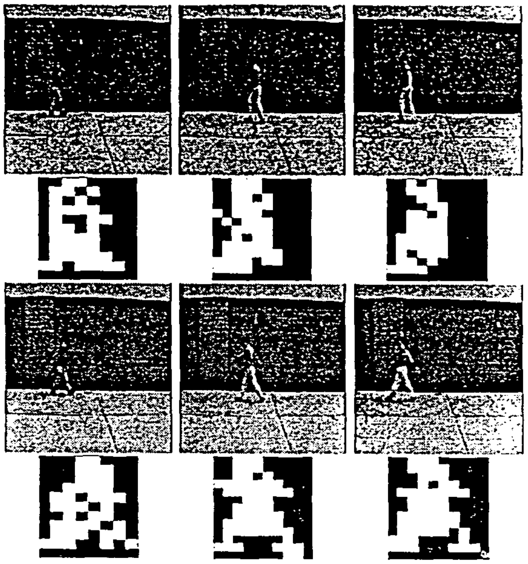


Fig. 6. First and third rows: Left column: Walking position from the training set, Middle and right columns: Resulting frames with the minimum matching costs. Second and fourth rows: DCT coefficient difference for the corresponding frames.

the rank information is preserved and they can be used without any decoding procedure. This method is fast since it does not require a fully decompressed MPEG video or JPEG image. The processing unit for the algorithm is a DCT block that is readily available from the compressed image.

Since the DC-DCT coefficients give the average intensity values of the blocks, one can get rid of the local luminance changes due to the reflection and other factors. Besides the processing speed, this method also smoothes the image to test the system performance for different resolution levels. Usually, the skin information from the DCT values of color components can not be used for human detection since the resolution requirement is not met. If the skin regions are detected (Fig. 7), the next step will be the segmentation and implementation of the proposed model based graph matching algorithm on the DC luminance blocks for each frame. The graph matching algorithm is explained in Section IV-D.

For low resolution and monochrome JPEG images, we propose a new algorithm for human detection. The system detects people in arbitrary positions in the image and in different scales. This approach is described in the next section.

C. Human Detection in Lower Resolution and Monochrome JPEG Images

In this new algorithm, the overall shape of a standing or walking person (from front or back-view) in still images is detected by using the AC-DCT coefficients. Most of the retrieval systems that are based on the compression schemes are devised for particular objects. Photobook [46] project uses a compact eigenspace representation of faces that can be used for both recognition as well as image compression. In Papageorgiou's work [51], the structural information of pedestrians is presented by a subset of wavelet coefficients and pedestrians are detected by the support vector machine classification

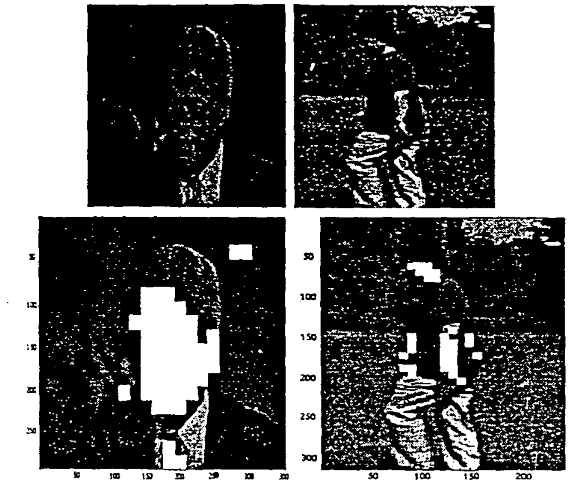


Fig. 7. Top: Original frames (YCbCr: 4:2:0 and 4:4:4), Bottom: Marked frames with macroblocks detected as skin regions.

method. Our work aims to retrieve information from images and videos compressed using standard algorithms such as JPEG and MPEG. This differentiates our approach from previous work where the compression algorithms are determined by characteristics of object of interest to be retrieved. The proposed algorithm is displayed in Fig. 8.

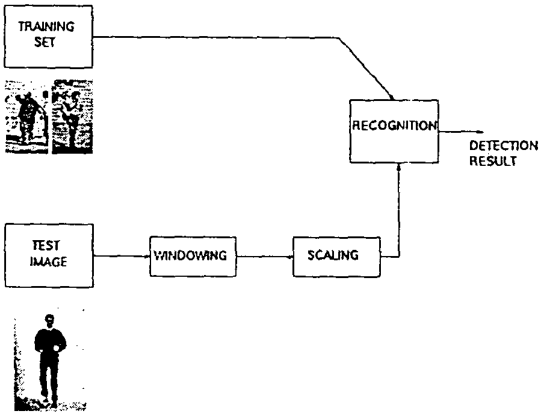


Fig. 8. Human detection system for low resolution JPEG images.

To capture the intensity variations, first order AC coefficients (c_{01}, c_{10}, c_{11}) are used (Fig. 10). DCT coefficient values capture the local directionality and coarseness of the spatial image. The vertical (horizontal) edges in uncompressed image correspond to high frequency component in the horizontal (vertical) frequencies and diagonal variations correspond to channel energies around the diagonal harmonics. Our approach is based on the observation that the structural information of human silhouettes can be captured from AC-DCT coefficients. In particular, energy of blocks that is obtained by summing up the absolute amplitudes of the first order harmonics is used. The sides of the human body have a high response to the vertical harmonics while AC coefficients of the horizontal harmonics capture head, shoulder and belt lines (Fig. 10). Furthermore, the corner edges at shoulders, hands and feet contribute to local diagonal harmonics.

To train our system, 800 pedestrian images obtained from the Artificial Intelligence Laboratory at MIT, are used. The pedestrians are centered in these 128x64 pixel windows. The windowing step in Fig. 8 determines a 128x64 window and shifts it throughout the test image. The regions that have a lower AC energy than a given threshold (uniform regions), are eliminated. The following step resizes the image part in the 128x64 window to achieve multiscale detection. The scaling operation is done in compressed domain [58]. Note that the computational complexity of the transform domain manipulation techniques strongly depends on the number of zero DCT coefficients. Since the proposed algorithm uses three AC coefficients, the required computation can be further reduced by using sparse matrix multiplication techniques or other fast schemes in transformed domain [59].

Our goal is to find a compact representation of the human silhouette by computing the principal components of the energy distribution of human bodies, or the eigenvectors of the covariance matrix of the human body images. These eigenvectors represent a set of features which together characterize the variation between human images. The number of eigenvectors (M) is equal to the number of images in the training set. In our algorithm we use the best eigenvectors ($M' = 12$) with the highest eigenvalues. Similarity measure in eigenspace representation for pattern matching in images is preserved under linear, orthogonal transformations. This implies that the principal component method gives exactly the same measure of match on transformed data as on pixel domain data. For lossy compression schemes such as JPEG and MPEG, the quantization of the transformed data is the cause for the degradation of the similarity measure. Although the DCT coefficients are quantized (furthermore, the coefficients except the three first order AC coefficients are quantized to zero), the essential information for matching purposes is preserved. The following steps summarize the recognition:

- Compute eigenvectors and eigenvalues from the training set of compressed human body images.
- Given an input image, calculate a set of weights based on the input image and the M' eigenvectors by projecting the input image onto each of the eigenvectors.
- Detect human regions by computing the distance between the mean adjusted input image and its projection onto human body space.

Let the training set of human images be $\Gamma_1, \Gamma_2, ..., \Gamma_M$, and the average be $\Phi = (\Gamma_1 + \Gamma_2 + ... + \Gamma_M)/M$. The difference of a human image from this average image is $\phi_i = \Gamma_i - \Phi$. Our goal is to find a set of M orthonormal vectors, $\mathbf{u_k}$ and their eigenvalues $\beta_\mathbf{k}$ which best describes the distribution of the data by using the principal component analysis. $\mathbf{u_k}$ and $\beta_\mathbf{k}$ are the eigenvectors and eigenvalues, respectively, of the covariance matrix C:

$$C = \frac{1}{M} \sum_{n=1}^{M} \phi_n \phi_n^T = AA^T \tag{6}$$

where the matrix $A = \frac{[\phi_1 \phi_2 ... \phi_M]}{\sqrt{M}}$. The matrix C is a N by N matrix and the calculation of eigenvectors and eigenvalues of this matrix is a difficult task. To reduce the computational complexity, the eigenvectors x_k and eigenvalues λ_k of the matrix A^TA are computed. It can be proven that the eigenvectors $\mathbf{u_k}$ of matrix C can be computed as [60]:

$$\mathbf{u_k} = \frac{\sum_{l=1}^{M} \phi_l x_{kl}}{\sqrt{\lambda_k}} \tag{7}$$

and the eigenvalues are the same those matching x_k. The first 12 eigenimages obtained from 800 training images are shown in Fig. 9. Creating the vector of weights for an image is equivalent to projecting the image onto the human body space. The distance ϵ between the image and its projection onto the body space is the distance between the mean adjusted input image $\phi = \Gamma - \Phi$ and $\phi_f = \sum_{k=1}^{M'} \omega_k \mathbf{u_k}$, its projection onto human body space, where $\omega_k = \mathbf{u_k^T}(\Gamma - \Phi)$ for $k = 1, ..., M'$.

The overall system performance was tested on 40 images. We achieve a correct detection rate of approximately 80%. The results are given in Section V. The system is also trained for background classification by using several images where human is not present.

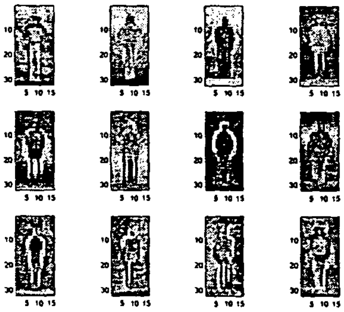


Fig. 9. 12 eigenimages (upsampled).

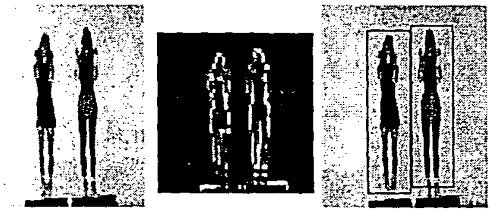


Fig. 10. Left: Original image, Middle: AC-DCT values, Right: Classification result.

IV. Human detection and posture recognition in uncompressed domain

This section describes the information retrieval from uncompressed images and videos. Furthermore, information obtained from the compressed domain processing techniques are used as a cue for further processing of the image/video in the uncompressed domain depending on the application and user needs. The arrows combining blocks A-B, A-C, and B-C in Fig. 1 indicate the information flow between these blocks.

21 oF 141

The following subsections describe the algorithm blocks illustrated in Fig. 2 that corresponds to Block C in Fig. 1 for the extraction of low level features from uncompressed images and videos or from the regions extracted in the compressed domain by using intensity, color and motion of pixels. Blocks C1, C2, C3, and C4 in Fig. 2 correspond to the segmentation, model based segmentation, object modeling by invariant shape attributes, and graph matching subsections, respectively.

A. Segmentation

Segmentation algorithms that use only low-level features fail in most cases due to the image noise, different illumination conditions, reflection and shadows. The solution for an automatic object segmentation is to manage the segmentation process by using object-based knowledge in order to group the regions according to a global constraint. In this work, a new model-based segmentation, where global consistency is provided by using the relations of pixel groups, is proposed. These groups are obtained from the combination or further segmentation of group results of a low level segmentation algorithm. Managing the segmentation process using a feedback from relational representation of the object improves the extraction result even if its interior or its boundary is changed partially.

Our overall segmentation algorithm has three steps. The first step entails moving object extraction for uncompressed video sequences. The extraction algorithm presented in this section is a modified version of Kanade-Lucas-Tomasi's tracking algorithm. The output of this algorithm is a set of rectangular regions including moving objects. The second step is color image segmentation combined with an edge detector where small segments are removed. The last segmentation step, curvature segmentation, helps to get the primitive segments by dividing the complex object parts into simpler ones. Resulting segments produced from this initial segmentation are combined by using a bottom-up control. We show that proposed model-based segmentation increases the overall algorithm performance by eliminating the segments that belong to the background.

The contribution of the overall segmentation algorithm is to use feedback from relational representation of the object to guide the segmentation process. We improve object extraction by reducing the dependence on the low level segmentation process and combining the boundary and region properties. Furthermore, the features used for segmentation (i.e. color, motion, curvature) are also attributes for object detection in relational graph representation. This property enables to adapt the segmentation thresholds by a model-based training system.

A.1 Motion Segmentation

This part corresponds to uncompressed video applications where moving objects are extracted. In a video sequence, the feature points of an object are tracked based on Kanade-Lucas-Tomasi tracking method [37].

A point (x, y) in the first image I moves to point (W_x, W_y) in the second image J, where:

$$J(W_x(x,y), W_y(x,y)) = I(x,y), \quad W_x(x,y) = \sum_p^n \sum_q^n a_{pq} x^p y^q$$

$$W_y(x,y) = \sum_p^n \sum_q^n b_{pq} x^p y^q \tag{8}$$

Given the successive frames I and J, the problem is to find the parameters in the deformation matrix W and d, where d $= [a_{00} \quad b_{00}]^T$. The problem is the choice of the parameters that minimize the dissimilarity ϵ.

$$\epsilon = \int\int_W [J(W_x(x,y), W_y(x,y)) - I(x,y)]^2 \mathrm{dxdy} \tag{9}$$

where W is the given feature window. After Taylor series expansion, d is determined by solving the equation Zd $=$ e where:

$$Z = \int\int_W \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} \mathrm{dxdy} \tag{10}$$

The eigenvalues of Z determine the selection of feature points, where d provides information about the displacement of the feature points in the second frame. The feature points with large eigenvalues correspond to high texture areas that can be matched reliably. These points are grouped according to their moving directions and distances (Fig. 11). Only the feature points with a velocity greater than a given threshold are considered. Next step is the determination of a rectangular region of interest by calculating the center of gravity and the eccentricity of these groups. If the area of this region is smaller than a threshold defined by the maximum object size in the frame, this region is not processed.

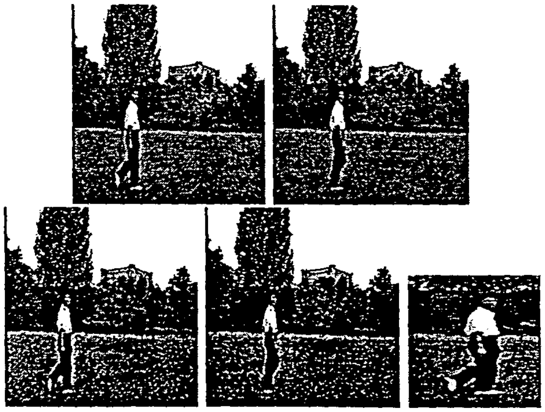


Fig. 11. Top: Initial and final video frames; Bottom Left and Middle: Tracked features (motion threshold = 1pixel/frame. distance threshold = 15 pixels): Bottom Right: Potential area that contains OOI.

A.2 Region Based Segmentation

An object usually contains several sub-objects; such as head, torso, limbs, etc. of a human, which can be obtained by segmenting the object hierarchically into its smaller unique parts. Here, the color image segmentation technique proposed in Harris [63] combined with an edge detector algorithm is used for rigid and non-rigid objects. For human detection, a skin color model is formed via Farnsworth nonlinear transformation.

The extraction of object of interest is a difficult task, especially in still images with a nonuniform background. As a result, the segmented image can contain regions corresponding to the background. However, these regions will not match to the regions of the template object. Semantic segments are created from the combination of low level edges or region based segments. If the object boundaries were segmented accurately, the shape descriptors for each object part could give satisfactory results for shape retrieval. However, a general automatic object segmentation without any user interface is almost impossible due to the illumination changes, shadows and occlusions especially for still images. Although using features invariant to illumination or reflection can improve the segmentation results, it is still not enough alone. Prior knowledge about the object to be retrieved should be used to segment the regions properly. One method is to perform rigid and deformable model based segmentations [64], [65], [66], [67]. The latter work differs from the previous works by enforcing global consistency. Local and global constraints should be used together for a segmentation that is robust to occlusions and variations in object shapes. These approaches try to extract the object boundary. Our approach differs from them at this point and will be explained in the next subsections.

A.3 Curvature Segmentation

The segmented region boundaries can still be in complex forms. The boundaries are first smoothed. Concave and convex segments (landmarks) that are used for curvature segmentation are determined on the resulting contour. The main reason for finding boundary landmarks is that they can be used to partition complex parts into simpler domains. For example, these landmarks are used to partition the arm into upper-arm and lower-arm. In this subsection, Gaussian based smoothing followed by curvature segmentation is studied. Gaussian smoothing is suitable for smooth human body parts in order to reduce the effect of image noise and clothing.

Gaussian Based Smoothing

The contour shape analysis is implemented to extract the convex parts of objects that determine visual parts separated by concavities. A method is to smooth of the boundaries by using a 1-D Gaussian kernel and then to calculate the curvature of each boundary point [11]. The width of the kernel defines the scale at which curvature is estimated. The noise and fine details are smoothed at large width, leaving distinct extrema at positions of perceptually significant points on the boundary. These points are called landmarks. Fig. 12 shows the Gaussian smoothing result for the human body part. As an example, the arm and leg segments are smoothed with a Gaussian kernel and the landmarks are defined.

Next step is the curvature segmentation regarding to these landmarks.

After the Gaussian smoothing operation, the concave points with high curvature K_s (greater than a threshold th_k) and arc lengths (greater than a threshold th_s relative to the segment length) are marked. A normal line is computed from this landmark until it reaches another point on the contour. Then, the segment is divided at these points and an interpolation is performed between these points to form closed segments. As expected, experimental results show that the high curvature locations occur at the joints on the limbs. Since human body parts are smooth objects the smoothing factor is chosen very small (= 1.25). Curvature threshold is chosen the same for all the test images (= 0.55) and arclength threshold is 20%. In Fig. 13, the curvature segmentation result for selected body parts is shown. Note that, since the arc length at the junction of the legs (belly) is small relative to the whole segment length, this part is not segmented. The graphs, given in Fig. 13, show the curvature points. For the arm segment, there is one concavity point which is greater than the curvature threshold while for the leg segment, all the concave points are below this threshold. Fig. 14 displays another example from a MPEG7 test sequence.

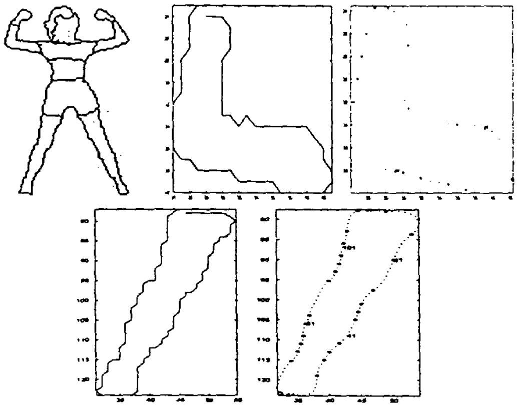


Fig. 12. Gaussian smoothing results for the arm and leg segments of the example human body with the landmarks.

Surface Approximation (Modeling by Superellipses)

Even when human body is not occluded by another object, limb positions may cause occlusion of body parts in different ways. For example, a hand can occlude some part of torso or legs. In this case the combination of occluded part with hand is not meaningful. However, 2D approximation of parts by fitting superellipses with shape preserving deformations provides more satisfactory results. It also helps to disregard the deformations due to the clothing. Result of the global approximation which do not capture local deformations seems more appropriate for human body. Hence. instead of using region pixels it is better to use parametric representations to compute shape descriptors. In a similar work by Bennamoun et al. [17]. a simple vision system where the objects are modeled by superellipses is proposed. Since their system performance highly depends on the initial segmentation results, they use single test objects with uniform backgrounds. The recognition stage compares the angles of the test object skeleton with the library object

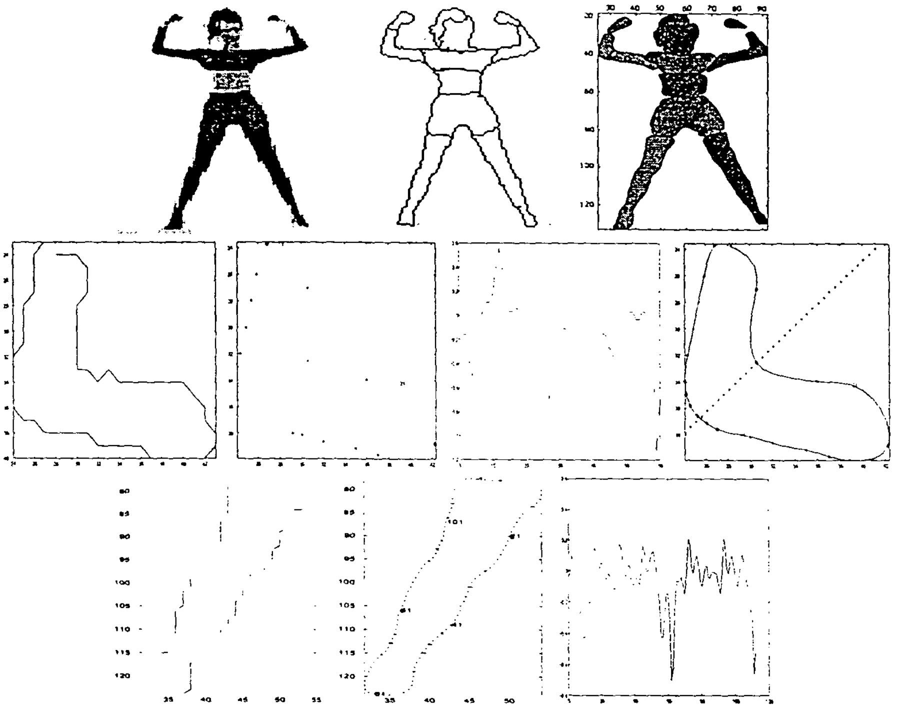


Fig. 13. Top: First: Original image. Second: Segmentation result. Third: Curvature segmentation results. Middle: First: Arm segment. Second: Smoothed contours with landmarks ($th_k = 0.55$). Third: Curvature points. Four: Curvature segmentation. Bottom: First: Leg segment. Second: Smoothed contours with landmarks ($th_k = 0.55$). Third: Curvature points.

skeleton and decides if the same object is present in the library. Their algorithm can only be used for non-occluded objects with a certain orientation, where our system can overcome the initial segmentation problem with the model based segmentation, can work for occluded images without any orientation constraint and combine the object parts via graph matching algorithm and decide the human presence. The detailed procedure for superellipse fitting is given below.

A superellipse can be described explicitly as:

$$x = f_x(\eta) = a_x cos(\eta)^\epsilon, \qquad y = f_y(\eta) = a_y sin(\eta)^\epsilon \tag{11}$$

In these equations, $-\pi < \eta < \pi$, a_x and a_y are two semi-axis, and ϵ is the roundness parameter. The curve intersects the x axis at a_x and $-a_x$ and intersects the y axis at a_y and $-a_y$. The inside-outside function of a two dimensional superquadric can be given as:

$$(\frac{x}{a_x})^{2/\epsilon} + (\frac{y}{a_y})^{2/\epsilon} = f(x, y, \mathrm{a}) \tag{12}$$

where a is the parameter set. There can be various deformations that can be implemented on the superellipses. Tapering and bending are sufficient deformations to represent human body. However, when for example legs are wide open they have to be segmented since no shape preserving deformation can represent them. Tapering along the y-axis is:

$$X = (\frac{K}{a_y} + 1)x, \qquad Y = y \tag{13}$$

where K is a constant. Circular bending:

$$X = x + sign(b)(\sqrt{y^2 + (a_y/b - x)^2} - (a_y/b - x))$$

$$Y = sin(atan(y/(a_y/b - x))(a_y/b - x) \tag{14}$$

In these equations, b is the bending parameter, (X, Y) are the deformed (x, y) values where $(D \circ R \circ T)(x, y) \rightarrow (X, Y)$ with D =Deformation, R =Rotation, T =Transformation. In order to find superellipse parameter set $\mathbf{a} = [a_x, a_y, \epsilon, K, b, \theta, p_x, p_y]$, that fits best to the segment data (X, Y), Levenberg-Marquardt method is used [68] for nonlinear parameter estimation. First, the initial parameter set is used to find non-deformed world centered superellipse $(\overline{x}, \overline{y})$ where $(D \circ R \circ T)^{-1}(X, Y) \rightarrow (\overline{x}, \overline{y})$

The model to be fitted, the inside-outside function $f(\overline{x}, \overline{y}, \mathrm{a})$ forms the merit function χ in order to determine best fit parameters by its minimization. With nonlinear dependences, the minimization must proceed iteratively. The procedure is repeated until χ^2 stops decreasing.

$$\chi^2(\mathrm{a}) \;=\; \sum_{i=1}^{N}(1 - f(\overline{x}, \overline{y}, \mathrm{a}))^2 \tag{15}$$

Some examples for superellipse fitting are displayed in Fig. 15.

B. Model Based Segmentation

The combination of features related to the boundary and interior of the object along with the relationships between the parts is more robust since the other one works when one fails. For this reason, the proposed method and segmentation procedure are implemented iteratively. Closed regions are defined and small ones are removed. For each segment and the combinations of these segments formed by merging them according to the adjacency information, the attributes (unary and binary), that are given in detail in graph matching section, are computed. For comparison of the test and model data, the graph matching algorithm is implemented and the number of regions, that are matched, is checked. If sufficient number of regions is matched the unmatched regions are removed and the object regions are extracted.

The drawback of this approach for on-line applications is the computation of the various combinations of regions to be merged. The idea is to merge the neighboring regions. However, the connectivity constraint alone is not sufficient since

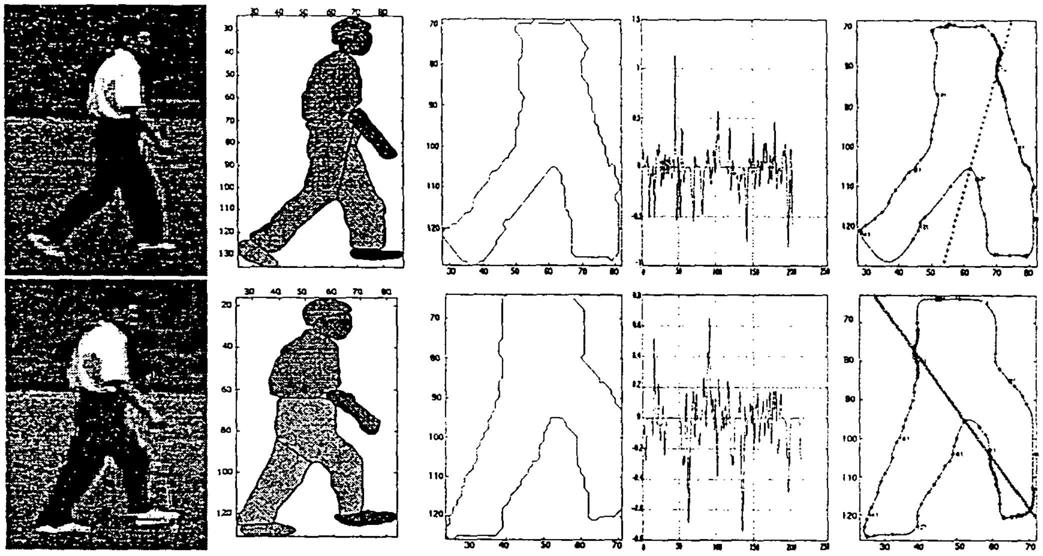


Fig. 14. First column: KLT algorithm result for the MPEG7 test sequence. Second column: Segmentation results. Third column: Leg segment. Fourth column: Curvature of the segment($th_k = 0.55$). Fifth column: Curvature segmentation.

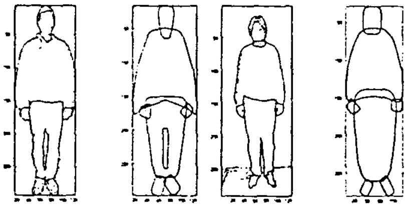


Fig. 15. Approximations for two bodies

testing all combinations to select the best match is impractical due to the computational complexity. This number will increase exponentially with the number of regions under consideration. For human images, a meaningful combination is the combination of adjacent segments on the same principle axis. For example, upper arm of a person with a shirt can be segmented into two parts, however it should be the combination of clothed and naked regions. The opposite of this example can also occur, and color and curvature segmentation can fail in extracting the desired object parts. For example, two adjacent object parts in the image might correspond to one node in the model image, e.g., color and curvature segmentation can fail to segment arms from torso. It is shown that this segmentation effect is removed by using possible combinations of the object parts.

C. Object Modeling by Invariant Shape Attributes

For object detection, it is necessary to select part attributes which are invariant to two dimensional transformations and are maximally discriminating between objects. Geometric descriptors for simple object segments which correspond to the vectors in the graph nodes such as area, circularity (compactness), weak perspective invariants [69], and spatial relationships are computed. These descriptors are classified into two groups: unary and binary features.

In order to obtain high level semantics, a relational graph, where each node of this graph corresponds to a segmented part with its feature vector and each arc to their relationship, is built. Matching of the relational graphs of objects with the reference model yields to the detection of objects. The aspect graph of the reference object is formed according to the segmentation results of the training images.

Since the object is composed into its primitive subparts, simple attributes revisited in this section are sufficient to describe the segments characteristics. Furthermore, the following extensions are done for application specific algorithms: Since detection of skin regions in color images greatly increases the performance of human detection an elaborate skin color model based on a perceptually uniform color space is formed. Relative position and orientation obtained from the weak perspective invariants are used to detect human articulated movements.

C.1 Unary Features

The unary features for human bodies are:

a) compactness; b) eccentricity; c) color (hair and skin).

The eccentricity is calculated as the ratio of length of the minor axis to the length of the major axis, which is also the ratios of the eigenvalues of the principal components. The circularity (compactness) of the region provides a measure of how close the region is to a circle. To represent the skin and hair color, perceptually uniform color system (UCS), proposed by Farnsworth [70] is used. Like other attributes, color attribute (c_j) of an image segment will be separated by a distance from the model color (c_i) with tristimulus values (t_1, t_2, t_3). This color difference measure must reflect noticeable color differences in order to capture skin and hair color models. First RGB color information is converted to XYZ color system and the resulting chromaticity components are transformed using Farnsworth nonlinear transformation to the new chromaticity (u, v) values. The noticeable color differences in the XY chromaticity diagram can be fitted by ellipses, but these color differences become much more circular and tend to be uniform in the UV diagram [70]. These (u, v) values and the luminance are used to determine skin and hair locations in the image with adjacency and shape attributes (Fig. 16). Our method relies mainly on the skin color model since the hair color model is not that reliable.

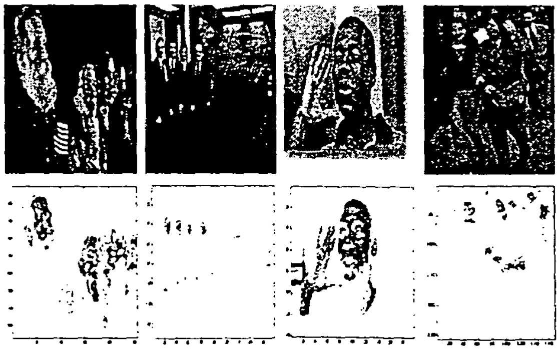


Fig. 16. Skin color segmentation results for some test images.

C.2 Binary Features

The binary features are: a) Ratio of areas; b) relative position and orientation; c) adjacency information between nodes with overlapping boundaries or areas. The relative position and orientation (Fig. 17) are computed using the weak perspective approximation [69]:

$$u = \frac{(\vec{p_3} - \vec{p_1}).(\vec{p_2} - \vec{p_1})}{|\vec{p_2} - \vec{p_1}|^2}, \qquad v = \frac{(\vec{p_3} - \vec{p_1}).(\vec{p_2} - \vec{p_1})^{\perp}}{|\vec{p_2} - \vec{p_1}|^2}$$

$$\cos(\alpha) = \frac{(\vec{p_2} - \vec{p_1}).(\vec{p_4} - \vec{p_3})}{|\vec{p_2} - \vec{p_1}||\vec{p_4} - \vec{p_3}|} \qquad (16)$$

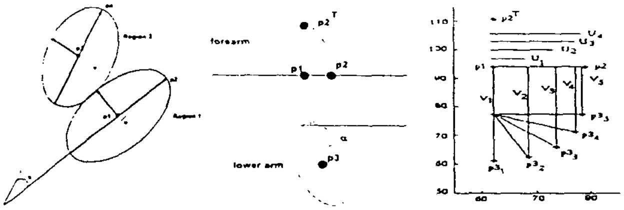


Fig. 17. Left: Relative position(RP) and orientation(OR) of two regions. Middle: Arm model. Right: RP and OR changes of the forearm and lower arm with respect to each other.

D. Graph Matching

Human is a complex object formed by several simple visual parts (head, torso, hands, etc.). The learning of the shape of OOI is then related to the learning of the organization of simple visual forms that make up OOI with different attributes and spatial relationships among themselves.

Consider a human recognition application where head, arms, legs and torso are segmented and described by a set of unary and binary features. A system that contains unary and binary classification mappings must also be able to interpret the match and check the conditional rules in order to index the parts correctly. Our solution to this problem is to store the graph representation of the objects.

Although graph matching is widely used for representation of complex objects and scenes [6], [71], [72] and has a long history, it faces problems mostly due to the dependence on the segmentation results. For instance, a graph representation system called Acronym [73] that has been tested on aerial images to classify airplanes, failed when the extracted airplane features were not close enough to expected ones.

To overcome this problem, a new model based segmentation, that combines the initial segments or segments them to smaller parts using a feedback from graph representation of the object, is proposed. The reference graph representations of the objects are trained from the low level processing results. Extracted features for human detection differ also due to the different articulated movements and clothing. A graph matching algorithm with Bayesian framework is developed where conditional risk is minimized at every node of the branch to minimize the error rate.

Object detection is achieved by matching the relational graphs of objects (S regions) with the reference model. Note that S is the number of regions found after the low-level segmentation process. The combination of these segments for human presentation creates N nodes ($N \geq S$). The input image graph O_n with N nodes and a reference graph (O_r with N_r nodes) are matched. The aspect graph of the reference object is formed according to the segmentation results of the training images. In order to determine the body parts under the assumption that the unary and binary (relational) features belonging to the corresponding parts are Gaussian distributed, multi-dimensional Bayes classification is used. The graph matching algorithm is described below.

D.1 Graph Matching Algorithm

Two reference models namely front and side view models for human are used in the experiments. Our assumption is that human face (at least a part of it) must be seen since skin color is a dominant attribute for head (Fig. 18). Face detection allows to start initial branches efficiently and reduces the complexity. B_h represents the group of branches for the corresponding head area. Note that false face detection will result in a branch with single or very few matched nodes and will be eliminated. Relational graph matching would allow human detection without face part however it would increase the computational complexity significantly and it is left for future work. Each body part and meaningful combinations represent a class (ω). The combination of binary and unary features is represented by a feature vector (X). Note that feature vector elements change according to body part and the nodes of the branch under consideration. For example, for the first node of the branch, feature vector consists of unary attributes. The feature vector of the following nodes includes also binary features dependent on the previous matched nodes in the branch. For the purpose of determining the class of these feature vectors a piecewise quadratic Bayesian classifier is used. In our case, it is a multiclass and multifeature problem. For the reference model supervised learning is implemented using several test images. The features for each body part are assumed to be Gaussian distributed.

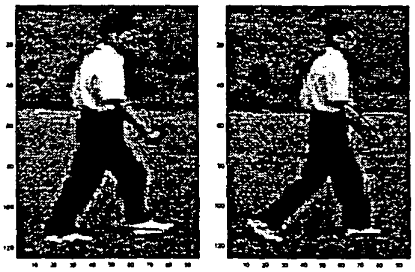


Fig. 18. Modeling detected skin parts with superellipses.

From Bayes theorem:

$$k = arg\max_j P(\omega_j|X) = \max_j \frac{p(X|\omega_j)P(\omega_j)}{p(X)} \rightarrow X \in \omega_k \qquad (17)$$

where $P(\omega_j)$ is a priori probability, $P(\omega_j|X)$ is a posteriori probability and ω represents a class. From [74], the discriminant function can be written as

$$g_j(X) = log(p(X|\omega_j)) + log(P(\omega_j)) \tag{18}$$

For multifeature problems with arbitrary covariance the decision surfaces are hyperquadrics and the resulting discriminant functions are

$$g_j(X) = X^T W_j X + \omega_j^T X + \omega_{j0} \tag{19}$$

In Eq. 19

$$W_j = -1/2\Sigma_j^{-1}, \qquad \omega_j = \Sigma_j^{-1} M_j,$$
$$\omega_{j0} = -1/2 M_j^T \Sigma_j^{-1} M_j - 1/2 log|\Sigma_j| + log P_{\omega_j}$$

where M_j represents the class mean and Σ_j is the covariance matrix of each class. During supervised learning, for each reference model node that represents a class $p(X|\omega_j)$ is computed. $P(\omega_j)$ is computed with the assumption that each class is equal probable and parts such as arms represent two classes in the model file. Note that our problem differs from the classical Bayes classification method in the sense that one does not try to find the class of a given feature vector by minimizing the risk factor but tries to find the existence of a member for a given class. Our goal is to detect OOI in the image by matching the image segments to possible classes of OOI. Due to the generality of the human detection problem and high variance of the within-class scatter matrices of unary feature vectors for different body parts, relational features must be used. Relational attributes explained in Section IV-C.2 are also elements of feature vector. Furthermore, conditional rule generation (r) eliminates the image segments that do not hold human body rules such as "face must be adjacent to torso", "if two arms are already matched in the branch there can not be another arm classification for that branch", and "angle between torso and face principal axis (α) can not exceed a certain threshold". Hence our problem is to find the existence of a member among image segments of a model class by maximizing the probability of feature vector for the given class in the corresponding branch. The overall algorithm for the relational graph matching is given below.

```
for every model node $j \in O_r$ do
    for every branch $b$ do
        $(i_1, i_2)$ =match$(j, b)$
        copy branch $b$ and add node pair $(j, i_1)$ in the
        new branch and update $G^b$ by adding $g_j^b(X_{i_1})$
        copy branch $b$ and add node pair $(j, i_2)$ in the
        new branch and update $G^b$ by adding $g_j^b(X_{i_2})$
        end for
end for
choose $arg\,\max_{b \in B_h} G^b$


match($j, b$)
for every image node $i \in O_n$ do
        for every matched node pair $(b_j, b_i)$ in the branch do
                if $\exists\, r(b_j, j)$ then
                        if $r(b_i, i)$ holds then
                                compute $g_j^b(X_{i,b_i})$
                        else
                                $g_j^b(X_{i,b_i}) = 0$
                        end if
                else
                                compute $g_j^b(X_i)$
                end if
        end for
end for
Return image nodes $i_1, i_2$ with two highest $g_j^b(x_i)$ values > threshold
```

V. Experimental Results

This section presents experimental results for human detection and posture and activity recognition in still images and video frames. The results for compressed and uncompressed domain techniques are given in the following subsections, respectively.

A. Compressed Domain

To evaluate the system performance for the activity recognition in compressed domain, several sequences with different activities are used. Table 1 displays the resulting normalized distances (Eq. 4) between the activity sets and test sequences. The results show that MPEG motion vectors corresponding to three human body subregions can be used for detection and recognition of human activity. Each test sequence gives the minimum normalized distance with its corresponding training set. The last sequence is a MPEG car movie. Note that the distances are very high for each activity class. Another restriction for car sequences is that the human body ratio is not suitable for the car mainbody. The performance of the algorithm depends on the temporal duration of the observed activity. The results displayed in the table are given for sequences with two or more activity periods. Results for low resolution and monochrome JPEG images are given in Fig. 19 where windows with distance ϵ values smaller than a predefined threshold are displayed .

Our results are compared with those of [51] for frontal and near-frontal poses since our system is trained only for these view angles. The authors in [51] use an overcomplete Haar dictionary of 16 x 16 pixels and train the system by using 564 positive examples that contain nonoccluded pedestrians and 597 negative examples that do not contain pedestrians. The detection rate for 141 nonoccluded pedestrian images in frontal or near-frontal images is 82%. In order to train our system, we use 800 positive examples and 600 negative examples with a bootstrapping algorithm. The test images contain a total of 126 non-occluded frontal poses and the algorithm can detect 101 of them correctly. Hence, we achieve

	Walking	Running	Kicking
walk1	0.001	0.0587	0.1543
walk2	0.0103	0.0929	0.0615
walk3	0.007	0.02	0.0784
walk4	0.0084	0.1218	0.1627
walk5	0.046	0.1506	0.1651
walk6	0.019	0.1298	0.208
run1	0.26677	0.0954	0.1688
run2	0.2525	0.0143	0.2519
run3	0.7665	0.027	0.1703
kick1	0.298	0.1253	0.0576
kick2	0.1901	0.109	0.0868
car	0.5362	0.4282	0.6922

TABLE I
THE NORMALIZED EUCLIDEAN DISTANCE BETWEEN THE ACTIVITY SETS AND TEST SEQUENCES.

a correct detection rate of approximately 80%. Our approach has the advantage of using the available data in standard compression algorithms and gives highly accurate detection results.

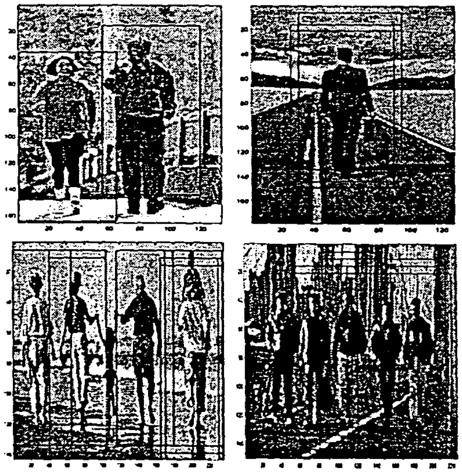


Fig. 19. Human classification results.

B. Uncompressed Domain

The performance of the proposed algorithm for non-rigid objects is given for 42 test images with human bodies for front and side views which are chosen from different sources. Since bending deformation increases the computational complexity, its value is set to zero and the computations are done using the tapering deformation. An example model file is shown in Fig. 20. In the model file, the adjacency information between parts is given as; head-torso, upper arm-torso, leg-foot, lower arm-hand, etc. For instance, there is no adjacency restriction between hand and leg or hand and belly, since hand can be at any position near them. In the model file these combinations are also chosen: arm = upper arm+lower arm, legs = leg1+leg2, lowbody = legs+belly, upbody = torso+belly, armtorso = arm+torso. Another important issue in the model file generation is that the features, such as eccentricity, can show large deviations from person to person (thin-fat, big-small, etc.) for each body part. Furthermore, eccentricity of the limbs are close to each other. Hence, within-class scatter matrix can be large while between-class scatter matrix can be small which is the worst case for a classification. Under the assumption that feature vectors have Gaussian distribution, their mean and variance are determined during supervised learning.

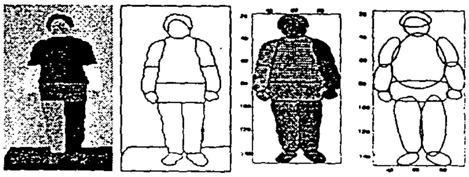


Fig. 20. First: The skin areas are determined in the model color image. Second: Segmentation result. Third: Curvature segmentation results. Four: Fitted superellipses to the body parts.

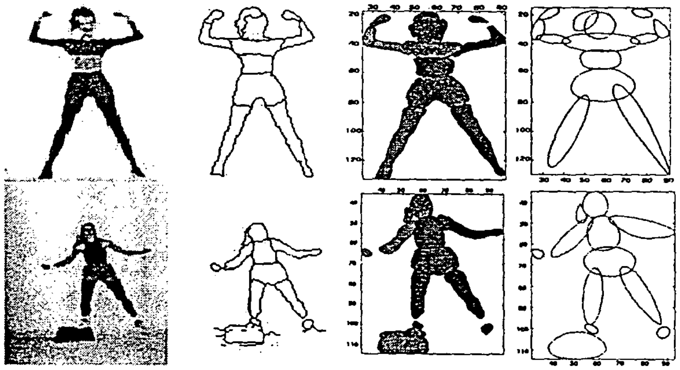
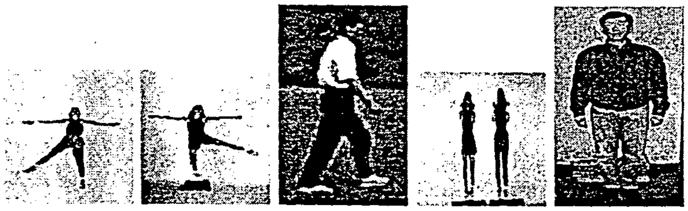


Fig. 21. Column 1: Original image. Column 2: Segmentation result. Column 3: Part separation and curvature segmentation results. Column 4: Fitted superellipses.



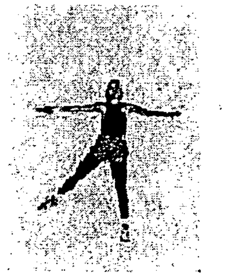
(a) (b) (c) (d) (e)

Fig. 22. Some test images. The detection performance for images a), d) and e) are given in Table 2

Results for segmentation and modeling with superellipses are displayed in Fig. 21 for different test images. After graph matching, the classification results for three images in Fig. 22 are given in Table 2. Note that, in Fig. 22 d), an image with multi-persons is tested. Since the algorithm first determines the face regions, two separate branches for each face region are initialized. In the same image, the lower arms of the persons are folded on their upper arms where graph matching algorithm classifies them as upper arms. The overall algorithm performance is obtained by computing the correct, false, and miss detection of the body parts in the test images. The preliminary results show that 70.27% of the body parts are correctly and 18.92% are falsely classified. The remaining 10.8% is the miss detection. In order to determine the posture of the persons in the still images and video sequences, the binary features of the corresponding matched node pairs are used after the classification. For example, the angle α between the image node matched to torso and image node (Section IV-C.2) matched to arm informs how much arms are open. Table 3 displays an example where both arms are open with an angle of 75-80 degrees, one leg is open with an angle of 40 degrees while other leg is approximately on the same axis as torso. Table 4 and 5 display the angles between torso1-arms and torso2-legs for the multi-person image. Since the angles are very small, it can be easily determined that both of the persons have closed arms and closed legs where their arms and legs are approximately on the same axis of torso. Note that, posture recognition is a direct result of correct classification of the body parts.

model - image(a)	model - image(d)	model - image(e)
face - face	face - face(Right body (r.b.))	face - face
torso - torso	torso - torso(r.b.)	torso - torso
belly - belly	belly - belly(r.b.)	legs - legs
arm1 - arm1	uparm1 - lowarm1(r.b.)	
arm2 - arm2	uparm2 - lowarm2(r.b.)	
leg1 - leg1	leg1 - leg1(r.b.)	
leg2 - leg2	leg2 - leg2(r.b.)	
	face - face(Left body (l.b.))	
	torso - torso(l.b.)	
	belly - belly(l.b.)	
	uparm1 - lowarm1(l.b.)	
	uparm2 - lowarm2(l.b.)	

TABLE II
CLASSIFICATION RESULTS FOR THREE TEST IMAGES.

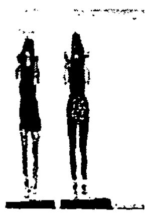


[!h]

Fig. 23. Test image.

part 1	part2	α
torso	arm 1	79.10
torso	arm 2	75.32
torso	leg 1	39.31
torso	leg 2	2.92

TABLE III
α VALUES ($\alpha = \Delta\theta$)



[!h]

Fig. 24. Test image.

part 1	part2	α
torso	arm 1	7.94
torso	arm 2	9.10
torso	leg 1	5.11
torso	leg 2	6.12

TABLE IV
α VALUES FOR THE LEFT BODY (FIG. 23).

part 1	part2	α
torso	arm 1	1.98
torso	arm 2	2.92
torso	leg 1	0.81
torso	leg 2	0.82

TABLE V
α VALUES FOR THE RIGHT BODY (FIG. 24).

VI. Conclusions

In this paper, we propose a hierarchical object-based image and video retrieval, specifically for human detection and activity recognition purposes. This work focuses in the problem of connecting low level features to high level semantics by developing relational object and activity presentations in both compressed and uncompressed domains.

The problem of object detection and activity recognition in compressed domain is addressed in order to reduce computational complexity and storage requirements. A new algorithm for object detection and activity recognition in JPEG images and MPEG videos is developed and we show that significant information can be obtained from the compressed domain in order to connect to high level semantics. Since our aim is to retrieve information from images and videos compressed using standard algorithms such as JPEG and MPEG, our approach differentiates from previous compressed domain object detection techniques where the compression algorithms are governed by characteristics of object of interest to be retrieved. An algorithm is developed using the principal component analysis of MPEG motion vectors to detect the human activities; namely, walking, running, and kicking. The algorithm is tested for sequences without camera motion. The distances of expansion coefficients between six sequences of walking people, three sequences of running people and two sequences of kicking people are presented to demonstrate that the classification among activities is clearly visible.

Object detection in JPEG compressed still images and MPEG I frames is achieved by using DC-DCT coefficients of the luminance and chrominance values. The performance is dependent on the resolution especially for human detection where skin region extraction is crucial. For lower resolution and monochrome images it is demonstrated that the structural information of human silhouettes can be captured from AC-DCT coefficients. In order to train our system, 800 positive (human) examples and different negative (non-human) examples with a bootstrapping algorithm are used. The overall system performance is tested on 40 images that contain a total of 126 non-occluded frontal poses and the algorithm can detect 101 of them correctly.

To increase the accuracy and to obtain more detailed information, the extraction of low level features from images and videos using intensity, color and motion of pixels and regions is done in uncompressed domain. Local consistency based on these features and geometrical characteristics of the regions is used to group object parts. The problem of managing the segmentation process is solved by a new approach that uses object based knowledge in order to group the regions according to a global consistency. A new model-based segmentation algorithm is introduced that uses a feedback from relational representation of the object. Object detection is achieved by matching the relational graphs of objects with the reference model. The algorithm maps the attributes, interprets the match and checks the conditional rules in order to index the parts correctly. The major advantages can be summarized as improving the object extraction by reducing the dependence on the low level segmentation process and combining the boundary and region properties. Furthermore, the features used for segmentation are also attributes for object detection in relational graph representation. This property enables to adapt the segmentation thresholds by a model-based training system. The detection rate for human body parts is 70.27% for images and sequences including human body regions at different resolutions and with different postures. The major contribution of the overall algorithm is to connect available data in compressed and uncompressed domain to high level semantics. The proposed hierarchical scheme enables working at different levels, from low complexity to low false rates.

In this paper. we propose a hierarchical human detection and activity recognition system in order to annotate databases for text based queries and to retrieve detailed information about the OOI. Our current work includes the study of the relationship between our algorithms proposed for human activity detection and the architectures required to perform these tasks in real time. For this purpose, we test the performance of the algorithm steps in terms of accuracy and computational complexity by using our testbed system with VLIW processors for video operations.

References

[1] I. B. Ozer, W. Wolf, A. N. Akansu, "Human activity detection in MPEG sequences," IEEE Workshop on Human Motion, pp. 61-66, 2000.

[2] I. B. Ozer, W. Wolf,"Human Detection in Compressed Domain," ICIP 2001, Thessaloniki, Greece, October 2001.

[3] I. B. Ozer, W. Wolf, A. N. Akansu,"Relational Graph Matching for Human Detection and Posture Recognition," SPIE, Photonic East 2000, Internet Multimedia Management Systems, Boston, November 2000.

[4] S. Loncaric, "A Survey of Shape Analysis Techniques," Pattern Recognition, vol. 31, no. 8, pp. 983-1001, 1998.

[5] S. Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, California, pp. 232-237, June 1998.

[6] R. M. Haralick and L. G. Shapiro, "Computer and Robot Vision," Addison Wesley Publishing Co., 1993.

[7] C. C. Chang, S. M. Hwang, and D. J. Buehrer "A Shape Recognition Scheme Based on Relative Distances of Feature Points from the Centroid," Pattern Recognition, vol. 24, pp. 1053-1063, 1991.

[8] A. Bengtsson and J. Eklundh, "Shape Representation by Multiscale Contour Approximation," IEEE PAMI, vol. 13, pp. 85-93, 1991.

[9] F. S. Cohen, Z. Huang, and Z. Yang, " Invariant Matching and Identification of Curves Using B-splines Curve Representation," IEEE Trans. on Image Processing, vol. 4, pp. 1-10, 1995.

[10] B. Gunsel and A. M. Tekalp, "Shape Similarity Matching for Query by Example," Pattern Recognition, vol. 31, no. 7, pp. 931-944, July 1998.

[11] A. P. Witkin, "Scale Space Filtering," Proc. 8th Int. Joint Conf. on Artificial Intelligence, pp. 1019-1022, 1983.

[12] F. Mokhtarian and A. K. Mackworth, "A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves," IEEE PAMI, vol. 14, pp. 789-805, 1992.

[13] M. K. Hu, "Visual Pattern Recognition by Moment Invariants," IRE Trans. Inform. Theory, vol. 8, pp. 179-187, 1962.

[14] F. Leymarie and M. D. Levine, "Simulating the Grassfire Transform Using an Active Contour Model," PAMI, vol. 14. pp. 56-75, 1992.

[15] A. H. Barr, "Superquadrics and Angle Preserving Deformations," IEEE Computer Graphics Applications, vol. 1, pp. 11-23, 1981.

[16] F. Solina and R. Bajcsy, "Recovery of parametric models from range images: the case for superquadrics with global deformations," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 12, no. 2, pp. 131-147, Feb. 1990.

[17] M. Bennamoun, B. Boashash. "A Structural-Description-Based Vision System for Automatic Object Recognition", IEEE Transactions on Systems, Man, and Cybernetics-Part B, Cybernetics, vol. 27, no. 6, pp. 893-906, Dec. 1997.

[18] J. R. Smith and S. F. Chang, "Querying by Color Regions Using the VisualSEEK Content-Based Visual Query System," Intelligent Multimedia Information Retrieval, Editor M. T. Maybury, AAAI/MIT Press, 1997.

[19] J. R. Smith and S. F. Chang. "VisualSEEk: A Fully Automated Content-Based Image Query System," Proc. ACM Multimedia Conf., pp. 87-98, Boston, 1996.

[20] H. Yu, W. Wolf, "A Visual Search System for Video and Image Databases," Proc. IEEE Multimedia, pp. 517-524, 1997.

[21] B. Moghaddam, H. Biermann. D. Margaritis, "Defining Image Content with Multiple Regions of Interest," CBAIVL, pp. 89-93, June 1999.

[22] E. Saber, A. M. Tekalp, "Integration of Color, Edge, shape, and Texture Features for Automatic Region-Based Image Annotation and Retrieval," ICIP, vol. 3, pp. 851-854, 1996.

[23] H. Wu, Q. Chen, and Y. Yachida, "Face Detection From Color Images Using a Fuzzy Pattern Matching Method." IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 21, no. 6, pp. 557-562, 1993.

[24] M. M. Yeung, B. L. Yeo, W. Wolf and B. Liu , "Video Browsing using Clustering and Scene Transitions on Compressed Sequences," SPIE vol. 2417 Multimedia Computing and Networking, pp. 399-413, 1995.

[25] A. M. Dawood and M. Ghanbari, "Scene Content Classification From Mpeg Coded Bit Streams," IEEE Workshop on Multimedia Signal Processing, pp. 253-258, 1999.

[26] H. Yu, W. Wolf, "Let's Video Freely- Automatic Video Indexing for Film and TV Program oriented Digital Video Library," SCI, pp. 217-222, 1998.

[27] Y. Yacoob and M. J. Black. "Parameterized Modeling and Recognition of Activities," ICCV, pp.120-127, 1998.

[28] J. K. Aggarwal and Q. Cai. "Human Motion Analysis: A Review," Computer Vision and Image Understanding. vol. 73, no. 3, pp. 428-440, March 1999.

[29] D. M. Gavrila. "The Visual Analysis of Human Movement: A Survey," Computer Vision and Image Understanding. vol. 73, no. 1, pp. 82-98, Jan. 1999.

[30] M. Walter, S. Gong, A. Psarrou, "Stochastic temporal Models of Human Activities," International Workshop on Modelling People, pp. 87-94, 1999.

[31] K. Rangarajan, W. Allen, M. Shah, "Matching Motion Trajectories Using Scale-Space", Pattern Recognition, vol. 26, no. 4, pp. 595-610, April 1993.

[32] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, W. von Seelen, "Walking Pedestrian Recognition", International Conference on Intelligent Transportation Systems, pp. 292-297, 1999.

[33] S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, " VideoQ: An Automatic Content-Based Video Search System Using Visual Cues," ACM Multimedia '97 Conference, Seattle, Nov. 1997.

[34] M. Kurokawa, T. Echigo, A. Tomita, J. Maeda, H. Miyamori and S. Iisaku, "Representation and Retrieval of Video Scene by Using Object Actions and their Spatio-temporal Relationships," ICIP, pp. 86-90, 1999.

[35] H. Miyamori, S. Iisaku, "Video Annotation for Content-based Retrieval using Human Behavior Analysis and Domain Knowledge," International Conference on Automatic Face and Gesture Recognition, pp 320-325, 2000.

[36] Y. P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge, "Rapid Estimation of Camera Motion from Compressed Video with Application to Video Annotation," IEEE Trans. on Circuits and Systems for Video Technology, vol. 10, no. 1, pp. 133-146, Feb. 2000.

[37] J. Shi and C. Tomasi, "Good Features to Track," CVPR, pp. 593-600, 1994.

[38] R. Jain, "Workshop Report: NSF Workshop on Visual Information Management Systems," Proc. SPIE Conf. on Vis. Commun. and Image Proc., 1993.

[39] R. Jain, A. Pentland, and D. Petkovic, "NSF-ARPA Workshop on Visual Information Management Systems," Cambridge, MA, June 1995.

[40] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafine, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," IEEE Computer, Vol 28, pp. 23-32, Sept. 1995.

[41] J. Dowe, "Content-based Retrieval in Multimedia Imaging," Proc. SPIE Conf. on Vis. Commun. and Image, 1993.

[42] S. Scraloff and A. Pentland, "Modal Matching for Correspondence and Recognition," IEEE Trans. Pattern Analysis Mach. Intell., vol. 17, pp. 545-561, 1995.

[43] S. F. Chang, W. Chen, and H. Sundaram, "Semantic Visual Templates - Linking Visual Features to Semantics," Proc. Int. Conf. on Image, pp. 531-535, 1998.

[44] A. Gupta, R. Jain, "Visual Information Retrieval," Communications of ACM, vol. 40, no. 5, pp. 70-79, May 1997.

[45] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, "Content-based image retrieval at the end of the early years," IEEE Transcations on Pattern Analysis and Machine Intelligence, Volume: 22, Issue: 12, pp. 1349-1380, Dec. 2000.

[46] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based Manipulation of Image Databases," International Journal of Computer Vision, 1996.

[47] V. E. Ogle and M. Stonebraker, "Chabot: Retrieval from a Relational Database of Images," Computer, vol. 28, no. 9, Sept. 1995.

[48] J. R. Smith and S. F. Chang, "Visually Searching the Web for Content," IEEE Multimedia, vol. 4, no. 3, pp. 12-20, July/Sept. 1997.

[49] W. S. Li and K. S. Candan, "SEMCOG: A Hybrid Object-based Image Database System and Its Modeling, Language, and Query Processing," Proceedings of the 14th International Conference on Data Engineering, pp. 284-291, Feb. 1998.

[50] U. Franke and D. Gavrila, "Autonomous Driving Goes Downtown," IEEE Intelligent Systems, vol. 13, no. 6, pp. 40-48, Nov. 1998.

[51] C. P. Papageorgiou, M. Oren and T. Poggio, "Pedestrian Detection Using Wavelet Templates," Proc. of CVPR, pp. 193-199, June 1997.

[52] S. F. Chang, J. R. Smith, M. Beigi, and A. B. Benitez, "Visual Information Retrieval from Large Distributed On-line Repositories," Communications of the ACM, vol. 40, no. 12, pp. 63-71, 1997.

[53] B. L. Yeo and B. Liu , "On the extraction of DC sequence from MPEG Compressed Video," ICIP, pp. 260-263, 1995.

[54] D. Schonfeld and D. Lelescu, "VORTEX: Video retrieval and tracking from compressed multimedia databases - template matching from MPEG2 video compressed standard," SPIE Conference on Multimedia and Archiving Systems III, Nov. 1998.

[55] Y. Zhong, H. Zhang, A. K. Jain, "Automatic Caption Localization in Compressed Video," IEEE PAMI, vol.22, no. 4, pp. 385-392, April 2000.

[56] H. Wang and S. F. Chang, "A Highly Efficient System for Automatic Face Region Detection in MPEG Video Sequences," IEEE Trans. on Circuits and Systems for Video Technology, special issue on Multimedia Systems and Technologies, vol. 7, no. 4, pp. 615-628, Aug. 1997.

[57] H. Wang, H. S. Stone, and S. F. Chang, "FaceTrack: Tracking and Summarizing Faces from Compressed Video," SPIE Multimedia Storage and Archiving Systems IV, 19-22 Sept., Boston.

[58] S. F. Chang and D. G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video," IEEE Journal on Selected Areas in Communications, vol. 13, no. 1, pp. 1-11, Jan. 1995.

[59] R. Dugad, N. Ahuja, "A fast scheme for downsampling and upsampling in the DCT domain," ICIP, pp 909-913, 1999.

[60] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," CVPR, pp. 586-591, 1991.

[61] M. Kirby and L. Sirovich, "Application of the Karhumen-Loeve Procedure for the Characterization of Human Faces," IEEE PAMI, vol. 12, no. 1, pp.103-108, 1990.

[62] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces," CVPR, pp. 586 -591, 1991.

[63] K. Harris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos, "Hybrid Image Segmentation Using Water Sheds and Fast Region Merging," IEEE Trans. on Image Processing, vol. 7, pp. 1684-1699, 1998.

[64] M. Nagao, T. Matsuyama, Y. Ikeda,"Region Extraction and Shape Analysis in Aerial Photographs," CGIP, pp. 195-223, 1979.

[65] M. Kass, A. P. Witkin and D. Terzopoulos. "Snakes: Active contour models," Inter. Journal on Comp. Vision, vol. 1, no. 4, pp. 321-331, 1988.

[66] H. S. Ip and D. Shen,"An Affine Invariant Active Contour Model for Model-Based Segmentation," IVC, pp. 135-146, 1998.

[67] L. Liu, S. Sclaroff,"Deformable Shape Detection and Description via Model-Based Region Grouping," CVPR, pp. 21-27, 1999.

[68] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical Recipes in C," Cambridge University Press, Second Edition, 1995.

[69] J. B. Burns, R. S. Weiss and E. M. Riseman, "View Variation of Point-Set and Line-Segment Features," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 15, no. 1, pp. 51-68, 1993.

[70] W. K. Pratt, "Digital Image Processing," J. Wiley and Sons, Second Edition, 1991.

[71] D. H. Ballard and C. M. Brown, "Computer Vision," Prentice-Hall, Englewood Cliffs, NJ, 1982.

[72] T. Caelli and W. F. Bischof, "Machine Learning and Image Interpretation," Plenum Press, New York, NY, 1997.

[73] R. A. Brooks, "Model-Based Three Dimensional Interpretations of Two Dimensional Images," PAMI, vol. 5, pp. 140-150, 1983.

[74] R. O. Duda, and P. E. Hart, "Pattern Classification and Scene Analysis," John Wiley and Sons, 1973.

Second Revised Manuscript JVCI2000-0003
A Graph Based Object Description for Information Retrieval in Digital Image and Video Libraries

I. Burak Özer[†], Wayne Wolf[†] and Ali N. Akansu[‡]

[†] *Department of Electrical Engineering,*
Princeton University
Princeton, NJ 08544, USA
[‡] *Department of Electrical and Computer Engineering, New Jersey Institute of Technology*
Newark, NJ 07102, USA

E-mail: iozer@ee.princeton.edu; wolf@ee.princeton.edu; ali@oak.njit.edu

Abbreviated Title: A Graph Based Object Description for Information Retrieval

Please send all correspondence to:

Dr. Wayne Wolf

Department of Electrical Engineering, Princeton University

Princeton, NJ 08544, USA

Tel: (609) 258-1424

Fax: (609) 258-3745

Email: wolf@ee.princeton.edu

This work focuses on the search of a sample object (car) in video sequences and images based on shape similarity. We form a new description for cars, using relational graphs in order to annotate the images where the object of interest (OOI) is present. Query by text can be performed afterwards to extract images of OOI from an automatically preprocessed database. The performance of the general retrieval systems is not satisfactory due to the gap between high level concepts and low level features. In this study we successfully fulfill this gap by using the graph based description scheme which provides an efficient way to obtain high level semantics from low level features. We investigate the full potential of the shape matching method based on relational graph of objects with respect to its accuracy, efficiency and scalability. We use hierarchical segmentation that increases the accuracy of the detection of the object in the transformed and occluded images. Many shape based similarity retrieval methods perform well if the initial segmentation is adequate, however, in most cases segmentation without a priori information or user interference yields unsuccessful object extraction results. Compared to other methods, the major advantage of the proposed method is its ability to create semantic segments automatically from the combination of low level edge or region based segments using model-based segmentation. It is shown that graph based description of the complex objects with model based segmentation is a powerful scheme for automatic annotation of images and videos.

Key Words: Object retrieval, graph matching, hierarchical car model. model based segmentation.

1. INTRODUCTION

With the rapid growth of multimedia information in forms of digital image and video libraries, there is an increasing need for intelligent database management tools. Although, the visual information is widely accessible, technology for extracting the useful information is still restricted. The traditional text-based query systems based on manual annotation process are impractical for today's large libraries requiring an efficient information retrieval system.

Multimedia information retrieval is a multidisciplinary area that is a combination of artificial intelligence, information retrieval, human interaction, and multimedia computing. It enables users to create, index, present, summarize, query, browse, and organize information within media such as text, audio, image, graphics, and video. Intelligent multimedia information retrieval includes those multimedia systems which go beyond hypertext environments. A significant amount of effort has been devoted recently to develop content-based retrieval systems where the images/videos are indexed by their intrinsic visual features.

Automatic annotation of images where an object of interest is present faces two major problems. One is the dependence of the object description on the feature extraction process which is a complex task especially for cluttered scenes. The other is that the visual properties of images that are described by feature vectors are difficult to describe automatically with text. Therefore, the similarity retrieval connecting these vectors to high level semantics and using high level knowledge to improve feature extraction become an important issue. In order to overcome these problems, in a previous paper [1], we have introduced an earlier version of the proposed method for retrieving images and video frames from a database where the content is modeled by a hierarchical system. The lowest level of information consists of pixels with color or brightness information. Features such as edges, corners, lines, curves and color/intensity regions are formed next. In the higher level, these features are combined to describe objects and their attributes. The low level features that form the object descriptors are connected to the high level semantics via a graph-based description scheme.

Another important issue in digital libraries is the query representation which is related to the user interface. Query by example (QBE) is a method of query specification that allows a user to specify a query condition by giving image examples, such as a photo of the object in the database that contains the shape to be retrieved. Main features of an image can be given as shape, spatial relation, color and texture. Another method is to draw the shape of the object. Sketch based retrieval is a special case of shape retrieval where the user describes a single object or a whole image by the layout of objects in it. Images are also retrieved by specifying colors and their spatial distribution in the image. Moreover, user can specify the movement of an object for video retrieval. If textual descriptions representing the content of images are available then a query by keyword can be performed. The proposed retrieval system is used for video sequences, images and sketches enabling text based queries.

The remaining of the paper is organized as follows. The following subsection covers related work where shape analysis techniques and retrieval systems are reviewed. Section 2 presents the proposed method where segmentation, shape descriptors and graph matching scheme are described in detail. The performance results are given in Section 3 while the last section includes the conclusions and future work.

1.1. Previous Work

Our description scheme is motivated by the well-known human perception theory and shape analysis techniques. The following subsections describe the related work.

1.1.1. Human Perception

Neural organization in the retina seems to be designed to provide information about the presence of discontinuities in the optical projection on the retina. It seems reasonable that the presence of borders, edges and contours in the stimulus would be the minimal information necessary for pattern perception since they could provide the building blocks for the perception of stable segregated portions of background and foreground [30]. Zusne [31] states the basic theories of visual form. The visual forms are transposable without loss of identity and will always be as good (regular, symmetric, simple, uniform) as the conditions allow. In our case, OOI (car) is the foreground object in an image or a moving object in a video frame. Car is a complex object formed by several simple visual parts (top and bottom parts of mainbody, windows, tires, etc.).

The importance of high curvature points for visual perception is emphasized by many researchers. Hoffman and Richards [32] investigated the significance of corners for perception. Their main point is that one can represent common objects by first indicating points at which contours change direction and secondly connecting appropriate ones with a straight line. Another remark is the fact that one can sketch the essence of a thing with a very few lines separated at corners. Thus polygonal approximation of a contour after eliminating small discontinuities provides the essentials of an object shape. The learning aspect of perception is studied by Hebb [33]. He states that the organization and mutual spatial relationship of object parts must be learned for successful recognition. The learning of the shape of OOI can then be related to the learning of the organization of simple visual forms that make up OOI with different attributes and spatial relationships among themselves.

1.1.2. Shape Retrieval

Many researchers have studied shape-based search. Shape based image retrieval is one of the hardest problems in general mainly due to the difficulty of segmenting objects of interest in images. The preprocessing algorithm determines the contour of an object depending on the application. Once the object is detected and located, its boundary can be found by using edge detection and boundary following algorithms [29]. However, the detection of the objects becomes a more difficult problem for complex scenes with busy background or many objects with occlusions and shading. Once the object border is determined its shape can be characterized by its shape features. These feature vectors are generated by using a shape description method to characterize a shape. The required properties of a shape description scheme are invariance to translation, scale, rotation, luminance, and robustness to partial occlusion. Afterwards, shape matching is used in model-based object recognition where a set of known model objects is compared to an unknown object detected in the image using a similarity metric.

Shape Analysis Techniques

Shape similarity methods can be classified into two parts, namely, contour and region based techniques [29]. Birchfield [34] stated that every closed set in a plane can be decomposed into its two disjoint sets; the boundary and the interior according to elementary set theory. Since these two sets are mathematically complementary, one can claim that the failure modes of a tracking module focusing on the object's boundary will be orthogonal to those of a module focusing on the object's interior. Since the same concept can be applied to shape analysis, the combination of contour and region based shape descriptors are used in the proposed system.

Contour-based Techniques:

For 1-D representation of shapes, Bennet and McDonald [35] use a tangent angle versus arc length function, that is also called turning function. The tangent angle at some point is measured relative to the tangent angle at the initial point. It is used by Arkin [36] for comparing polygonal shapes. The total turn (global curvature) is used for digital arcs by Latecki [39]. A signature of a boundary may be generated by computing the distance from the centroid to the boundary as a function of angle. Chang [40] constructs the distance function from the centroid to the feature points that are the points of high curvature. Template matching [27], chain coding [41], Fourier transform [42] and line segment moments [43] are other 1-D shape descriptors.

Scale space techniques rely on the object representation at different scales. Witkin [47] proposes a scale space filtering approach which provides a useful representation for significant features of an object filtered by low-pass Gaussian filters of variable variance. Asada and Brady [48] introduce a new representation called the curvature primal sketch that is obtained by computing curvatures at different scales. Mokhtarian and Mackworth [49] use the scale space approach as a hierarchical shape descriptor.

Another boundary representation technique is the curve approximation by utilizing polygonal and spline approximations. Polygonal approximations are used to approximate the shape boundary using the polygonal line. This is performed by using split-and-merge techniques based on some criteria. One approach is to merge points to form lines until exceeding a threshold. Bengston and Eklundh [44] propose a hierarchical method where the shape boundary is represented by a polygonal approximation. Splines have been very popular for the interpolation of functions and the approximation of curves. They possess the beneficial property of minimizing curvature [45, 46]. Latecki [50] uses an approximation of the segment contours in order to distinguish perceptually similar shapes. The main advantage of discrete contour evolution is that it does not cause shape rounding as in the case of Gaussian blurring.

The major drawback of these techniques is the dependency on the extraction of the object boundary. Another problem is the difficulty to evaluate the similarity between the boundaries of objects with high within-class variance.

Region-based Techniques

2-D moment based methods are among the most popular ones for regional descriptors [51]. The use of moments for shape description was proposed by Hu [52] who showed that moment based shape description is information preserving. An alternative transform approach is the Fourier transform of the shape. One of the disadvantages of these descriptors is that they do not reflect local shape changes. High-order features are required for shape classification. However, they are not robust to noise unlike low-order descriptors and are also computationally intensive. Surface approximation is another technique for region-based representation, e.g., superquadrics are widely used for modeling two and three dimensional objects in computer vision literature (Barr [16] and Bajcsy [19]). Medial axis transform first proposed by Blum [53] extracts a skeletal figure from the object and uses it to represent a shape by using a graph [54]. Leymarie and Levine [55] find the medial axis transform using snakes for active contour representation, high curvature points on the boundary, and symmetric axis transform.

As in the contour-based modeling, the performance of these techniques depend on the extraction of the object regions. Furthermore, higher order shape metrics is needed for the presentation of the complex objects. One solution is to decompose the object for its presentation as a combination of component shapes. For instance, if the object is decomposed to its simpler forms, the low order moment invariants can be used for subparts of the object. Furthermore, the result will be unaffected by a partial occlusion of the object. Some other simple region-based shape descriptors can then be used for these simpler forms, e.g., compactness (circularity), eccentricity of the region. Our research is motivated by the fact that shape analysis techniques can be effectively used if extraction of low-level features are governed by high level semantics.

1.1.3. Retrieval Systems

Content based image/video indexing and retrieval have been researched by the governmental [2, 3] and industrial [4, 5] groups as well as at the universities [6, 7, 8, 9, 10, 11, 12, 13]. They use different techniques based on image features such as shape, color, texture, motion or a combination of them. A survey of these retrieval systems can be found in Yoshitaka [14], Gupta [15], Rui [17] and Smeulders [18]. Some of these systems, described below, support query by keyword representing a semantic concept.

One of the systems is the Photobook [20, 21] which is a software tool for performing queries on image databases based on image content and textual annotation. It basically compares features associated with images. The content descriptions, where one is dependent on appearance, another uses shape, and the third one is based on textural properties; are combined with each other and with text based descriptions.

Cypress-Chabot [22] integrates the use of stored text and other data types with content-based analysis of images to perform "concept queries". In Webseek [23], the images and video are analyzed using visual features (such as color histograms and color regions) and the associated text utilized to classify the images into subject classes.

SEMCOG [24] system performs a semiautomatic object recognition. SEMCOG (SEMantics and COGnition-based image retrieval) aims at integrating semantics and cognition-based approaches to give users a greater flexibility to pose queries. COIR (Content-Oriented Image Retrieval), an object-based image retrieval engine based on colors and shapes is used. The main task of the COIR is to identify distinct image regions based on preextracted image metadata, colors and shapes. Since an object may consist of multiple image regions, COIR consults to the image component catalog for matching image objects.

One of the commercial systems is QBIC [4], which supports several basic image similarity measures such as average color, color histogram, color layout, shape and texture. QBIC is a research prototype image retrieval system that uses the content of images as the basis of queries. Queries are posed graphically/visually, by drawing, sketching or by keywords.

"Car" is one of the major objects of interest to be retrieved in the content-based retrieval systems. The systems, given below, cover different applications based on the extraction of car objects. Previous work on shape analysis of car objects was mostly based on boundary representation of objects. Dubuisson et al. [25] propose a segmentation algorithm using deformable template contour models to segment a vehicle of interest. Their goal is to determine the average travel time between two points in a road network by matching vehicles based on their color and shape attributes. The authors use five side view vehicle templates for classifying vehicle shapes where these templates are tested only for the side view car images with a stationary background. The results show that the classification depends on the detected edges. In another similar work [26], a vehicle is matched with a previously observed vehicle using color and shape features. Jain [27] proposes and tests a two dimensional shape matching and similarity ranking of still objects by means of a modal representation for car images. They employ selected boundary/contour points of the object with a coarse-to-fine shape representation. The algorithm is based on the contour detection of OOI. Papageorgiou et al. [37] use an overcomplete dictionary of Haar wavelets for identification of frontal and rear views of car in static images. Their method is based on the authors' previous work [38]. The approach is sensitive to partial occlusion. Xu [28] uses a hierarchical content description scheme and a hierarchical content matching technique for object retrieval. Experimental results are shown for a collection of car images. The authors form the root of the tree by grouping pixels similar in color therefore restricting the concept of homogeneity to color. Our approach is based on the observation that although complex objects can have shape and color variability within subparts of different objects, the relation between the subparts and the primitive shape characteristics are highly preserved. Therefore, our homogeneity concept is not limited to color.

2. PROPOSED SYSTEM

The proposed system is outlined in Figure 1 where an overview of each algorithm block is given in the following.

- **Image Preprocessing (B1):** In order to decrease the effect of high illumination changes due to different lighting conditions, a homomorphic filtering operator is applied.

- **Object Extraction (B2):** Separation of a moving object in a video (or foreground object in an image) is performed in this step. In a video sequence, we track the feature points of an object using the Kanade-Lucas-Tomasi tracking method [56] and group them according to their moving directions and distances (Figure 2). Only the feature points with a velocity greater than a given threshold are considered. Next step is the determination of a rectangular region of interest by calculating the center of gravity and the eccentricity of these groups. If the area of this region is smaller than a threshold defined by the maximum object size in the frame, this region is not processed. The output of this step is a rectangular region with an object of interest. For still images, we assume that our target still images have a foreground object at the front center of the image.

- **Object Segmentation (B3-B6):** An object usually contains several sub-objects (in our experiments, tires, windows, etc. of a car) that can be obtained by segmenting the OOI hierarchically into its smaller unique parts. Two different segmentation algorithms are implemented in this study. We use the color image segmentation technique proposed in [57] combined with an edge detector algorithm. Since it is not always possible to obtain a satisfactory result by using low-level segmentation algorithms, semantic segments are created from the combination of low level edges or region based segments. The details of the model-based segmentation (B6 in Figure 1) algorithm are given in the segmentation subsection (subsection 2.1.).

The segmented region boundaries can still be in complex forms. The boundaries are first smoothed by a polygonal approximation. The concave and convex segments (landmarks) on the resulting polygon are determined. The concavity/convexity information, the length and angle of line segments at the connection points are computed (subsection 2.1.3). These boundary landmarks can be used to partition complex parts into different domains, e.g., one can also isolate the mirrors from the rest of the body of a car. Furthermore, two adjacent object parts in the image might correspond to one node in the model image. It is shown that this segmentation effect is removed by using possible combinations of the object parts of the input image and model graph.

- **Object Modeling (B4):** Shape descriptors for simple object segments such as area, perimeter, circularity(compactness), moment invariants, weak perspective invariants [58] and spatial relationships are computed in this step. These descriptors are classified into two groups, namely, unary and binary features. For each segment of OOI, we assign these descriptors where unary features give information about the shape of a segment such as moment invariants and binary features provide information about the relationship between these segments such as adjacency information. These descriptors are described in subsection 2.2.

- **Relational Graph Matching (B5):** Graph matching is studied in [59, 60, 61]. In order to obtain high level semantics, we build a relational graph. Each node and arc of this graph corresponds to a vector for each segmented part including the geometric and relational features mentioned above (B4 in Figure 1). Matching of the relational graphs for objects with the reference model yields to the detection of objects that can be transformed and occluded in the database.

2.1. Motion and Region Based Segmentation

The purpose of image segmentation is to group pixels into regions that belong to the same object or object parts based on image homogeneity, e.g., color, texture, motion. An object can be found from appropriate grouping of object parts represented after a proper segmentation. Recognition is achieved by using the attributes of these groups. However, segmentation algorithms using only low-level features fail in most case due to the image noise, different illumination conditions, reflection and shadows. Although, some approaches use image features invariant to these conditions and improve the grouping results, they are not sufficient to detect complex objects. Consider to implement such an improved segmentation algorithm to segment a truck with a colored advertisement. It can only group regions of local consistency. The solution for an automatic object segmentation is to manage the segmentation process by using object-based knowledge in order to group the regions according to a global constraint. In this paper, a new model-based segmentation where global consistency is provided by using the relations of pixel groups is proposed. These groups are obtained from the combination or further segmentation of group results of a low level segmentation algorithm. Managing the segmentation process using a feedback from relational representation of the object improves the extraction result even if its interior or its boundary is changed partially.

Our overall segmentation algorithm has three steps. The first step is the extraction of moving objects from video sequences. The extraction algorithm presented here is a modified version of Kanade-Lucas-Tomasi's tracking algorithm. Output of this algorithm is a set of rectangular regions including moving objects where rest of the segmentation is implemented only in these bounding boxes. The second step is the object segmentation process. Color image segmentation is combined with an edge detector where small segments are removed. The third step combines resulting segments produced from this initial segmentation by using a bottom-up control. We show that proposed model-based segmentation increases the overall algorithm performance by eliminating the segments that belong to the background. Contour approximation for rigid objects reduces the noise effects on the rigid object segments.

The contribution of the overall segmentation algorithm can be seen in guiding the segmentation process using a feedback from relational representation of the object. The major advantages can be summarized as improving the object extraction by reducing the dependence on the low level segmentation process and combining the boundary and region properties. Furthermore, the features used for segmentation (i.e. color, motion, curvature) are also attributes for object detection in relational graph representation. This property enables to adapt the segmentation thresholds by a model-based training system.

2.1.1. Motion Segmentation

This part corresponds to video applications where moving objects are extracted. In a video sequence, the feature points of an object are tracked based on Kanade-Lucas-Tomasi tracking method [56].

A point (x, y) in the first image I moves to point (W_x, W_y) in the second image J, where:

$$J(W_x(x,y), W_y(x,y)) = I(x,y) \tag{1}$$

$$W_x(x,y) = \sum_p^n \sum_q^n a_{pq} x^p y^q \tag{2}$$

$$W_y(x,y) = \sum_p^n \sum_q^n b_{pq} x^p y^q \tag{3}$$

Given the successive frames I and J, the problem is to find the parameters in the deformation matrix W and d, where $\mathrm{d} = [a_{00} \quad b_{00}]^T$. The problem is the choice of the parameters that minimize the dissimilarity ϵ.

$$\epsilon = \int\int_W [J(W_x(x,y), W_y(x,y)) - I(x,y)]^2 \mathrm{dxdy} \tag{4}$$

where W is the given feature window. After Taylor series expansion, **d** is determined by solving the equation $Z\mathbf{d} = \mathbf{e}$ where:

$$Z = \int\int_W \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} \mathrm{dxdy} \tag{5}$$

The eigenvalues of Z determine the selection of feature points, where d provides information about the displacement of the feature points in the second frame. The feature points with large eigenvalues correspond to high texture areas that can be matched reliably. In addition to texture properties, we determine the good features to track according to their group properties. These points are grouped according to their moving directions and distances (Figure 2). Only the feature points with a velocity greater than a given threshold are considered. Next step is the determination of a rectangular region of interest by calculating the center of gravity and the eccentricity of these tracked feature groups. If the area of this region is smaller than a threshold defined by the maximum segment size in the frame, this region is not processed.

2.1.2. Region Based Segmentation with Model Based Segmentation

An object usually contains several sub-objects; such as tires, windows, etc. of a car, which can be obtained by segmenting the OOI hierarchically into its smaller unique parts (Figure 3). In this step, the color image segmentation technique proposed in Harris [57] combined with an edge detector algorithm is used for rigid objects.

The extraction of object of interest is a difficult task, especially in still images with a nonuniform background. As a result, the segmented image can contain regions corresponding to the background. However, these regions will not match to the regions of the template object. If the object boundaries were segmented accurately, the shape descriptors for each object part could give satisfactory results for shape retrieval. However, a general automatic object segmentation without any user interface is almost impossible due to the illumination changes, shadows and occlusions especially for still images. Although using features invariant to illumination or reflection can improve the segmentation results, it is still not enough alone. For a global consistency, semantic segments are created from the combination of low level edges or region based segments. For this purpose, prior knowledge about the object to be retrieved should be used to segment the regions properly. One method is to perform rigid and deformable model based segmentations [63, 64, 65, 66]. The latter work differs from the previous works by enforcing global consistency. Local and global constraints should be used together for a segmentation that is robust to occlusions and variations in object shapes. These approaches try to extract the object boundary. Our approach differs from them at this point and will be explained in the next sub-section.

Proposed Model Based Segmentation

The combination of features related to the boundary and interior of the object along with the relationships between the parts is more robust since the other one works when one fails. For this reason, the proposed method and segmentation procedure are implemented iteratively. Closed regions are defined and small ones are removed. For each segment and the combinations of these segments formed by merging them according to the adjacency information, the unary and binary attributes (subsection 2.2.) are computed. For comparison of the test and model data, the graph matching algorithm is implemented and the number of regions, that are matched, is checked. If sufficient number of regions is matched the unmatched regions are removed and the object regions are extracted.

Note that the combination of segments are matched to the model nodes with the evolution of graph matching, i.e., the unary attributes and the binary attributes related to the previous leaves of the branches for the combined segments are computed and compared to the model-based values for the possible matches. Hence, the low-level segments that are combined form high-level annotated segments via feedback from graph-matching evolution that is governed by the model-based knowledge. Figure 1 displays the relation where initial segments and the candidate nodes for the corresponding branch with previously matched nodes are the input to the model-based segmentation block where combined segments form new nodes. The attributes for these new nodes are then computed and are matched to the model graph nodes for possible annotations. These annotated segments are then used to form the subsequent leaves of the branches by means of the binary attributes.

The drawback of this approach for on-line applications is the computation of the various combinations of regions to be merged. The idea is to merge the neighboring regions. However, the connectivity constraint alone is not computationally efficient since testing all combinations to select the best match is impractical due to the computational complexity. Assume that one has four regions with the following structure; region 1 is a neighbor of region 2, region 2 is a neighbor of region 3 and 3, and region 3 is a neighbor of regions 2 and 4. The possible combinations are (1,2), (1,2,3), (1,2,3,4), (2,3), (2,3,4), (3,4) (Figure 5). This number will increase exponentially with the number of regions under consideration. One way to handle this is to constrain the color difference between regions. However, although the natural object color does not change significantly (e.g., skin, fruits, animals, trees) it is not true for objects such as cars. A part of a car can have several color components e.g., the mainbody of a car can be multi-colored. Best-first, or highest confidence first algorithms decrease the complexity [66] but also degrade the performance.

An example is displayed in Figure 4 for the car in Figure 3. The segmented region boundaries can still be in complex forms. The boundaries are first smoothed. Concave and convex segments (landmarks) are determined on the resulting contour (Figure 6). Contour approximation is explained more detailed in the following subsection.

2.1.3. Contour Evolution

In this method, digital curves which are composed of digital line segments are used. The idea is to decompose the digital curve into maximal digital line segments. In every evolution step, two consecutive line segments are replaced with a single line segment [50]. If the evolution is continued, the curve shape will be simplified. For each line segment pair the cost function is calculated and consecutive line segments with the minimum cost function are replaced with a single one. For each adjacent line segment pair $s1$ and $s2$, the cost function $K(s1, s2)$, which represents the significance of the contribution of arc $s1 \cup s2$ to the shape of digital curve C, is determined (Figure 7) using Eq. (6):

$$K(s_1, s_2) = \frac{\beta(s_1, s_2)l(s_1)l(s_2)}{l(s_1) + l(s_2)} \tag{6}$$

In Eq. (6), l is the length function normalized with respect to C. $s_1 = \overline{ab}$ and $s_2 = \overline{bc}$ are the two adjacent line segments in the decomposition of curve C, so that b is their common edge point and $\beta = \beta(s_1, s_2)$ is the turn angle. If the cost function is above the threshold the line pair $s1$ and $s2$ are replaced by the single line $\overline{ac}$. This process continues until the cost function for each pair is above the threshold. In our experiments the threshold is 0.01. It is argued that parts are generally defined to be convex or nearly convex shapes separated from the rest of the object at concavity extrema. The length of these concave and convex line segments as well as the angle between the corresponding lines (turn angle) are used as descriptors.

As reviewed in subsection 1.1.2., the main contour approximation methods are object filtering by low-pass Gaussian filters and curve approximation by using split-and-merge techniques. For rigid objects, we use this latter approach since it preserves better the characteristics of object parts. An example for the discrete curve evolution is displayed in Figure 8. Notice that the localization is not preserved in Figure 9 as in Figure 8 due to the global, independent Gaussian smoothing of the spatial components in the former case. The concavity measures that are computed from the normalized length and angle of the concave axes are displayed in Figures 10 and 11 for sketches and real images, respectively. The highest concavity points correspond to the landmarks for the two main subparts of the mainbody. Note that this feature can also be used for the ranking purposes, e.g. sport cars with hatchback have one maximum concavity point while sedan type cars have two main concavity points of a similar order. Other major concave axes on the mainbody correspond to the location of tires where adjacent concavities with a similar cost function K are observed.

2.2. Shape Descriptors

For object detection, it is necessary to select part attributes which are invariant to two dimensional transformations and are maximally discriminating between objects. Geometric descriptors for simple object segments, which correspond to the vectors in the graph nodes, such as area, circularity (compactness), weak perspective invariants [58], and spatial relationships are computed. These descriptors are classified into two groups: unary and binary features.

In order to obtain high level semantics, a relational graph is built. Each node of this graph corresponds to a segmented part with its feature vector and each arc to their relationship. Matching of the relational graphs of objects with the reference model yields to the detection of objects. The aspect graph of the reference object is formed according to the segmentation results of the training images. Since the object is composed into its primitive subparts, simple attributes revisited in this section are sufficient to describe the segments characteristics.

2.2.1. Unary Features

The unary features for rigid objects are:

a) Hu moment invariants; b) compactness (circularity); c) eccentricity; d)boundary shape code (turnangle and length of concave axes).

Moment invariants are defined in [52]. The basic idea of moment invariants is to define a set of measures which are invariant to scale, rotation, and translation changes in a 2D plane. Given a 2D intensity distribution $f(x, y)$, the moments of this function are defined as:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\mathrm{dxdy} \qquad \text{for p, q = 0, 1, 2, ...}$$

These invariants can be modified to include translational invariance in the following way:

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\overline{x})^p (y-\overline{y})^q f(x,y)\mathrm{dxdy}$$

where $\overline{x} = \frac{m_{10}}{m_{00}}$, and $\overline{y} = \frac{m_{01}}{m_{00}}$. Scale invariant moments can be derived from the above to give a set of normalized central moments:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \qquad \text{where} \qquad \gamma = \frac{p+q}{2} + 1$$

A set of 7 functions can be defined which are invariant to translation, rotation, and scale changes in the image plane [52]:

$$\begin{aligned}
\phi(1) &= \eta_{20} + \eta_{02} \\
\phi(2) &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi(3) &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi(4) &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi(5) &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{03} + \eta_{21})^2] \\
&\quad +(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi(6) &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{12} + \eta_{30})(\eta_{21} + \eta_{03}) \\
\phi(7) &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{03} + \eta_{21})^2] \\
&\quad +(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned}$$

The eccentricity is calculated as the ratio of length of the minor axis to the length of the major axis, which is also the ratios of the eigenvalues of the principal components. The circularity (compactness) of the region provides a measure of how close the region is to a circle. The boundary shape code includes the turnangle and length of concave axes. This attribute can be used for ranking purposes. For example, the shape code of a sedan car body differs from the shape code of a sport car body.

Eccentricity and circularity are defined as

$$\text{Eccentricity} \equiv \frac{\eta_{20} + \eta_{02} + \sqrt{(\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2}}{\eta_{20} + \eta_{02} - \sqrt{(\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2}} \qquad \text{Circularity} \equiv \frac{\text{Perimeter}^2}{4\pi \text{Area}}$$

2.2.2. Binary Features

The binary features are: a) Ratio of areas; b) Relative position and orientation; c) Adjacency information between nodes with overlapping boundaries or areas. The relative position and orientation (Figure 12) are computed using the weak perspective approximation [58]:

$$u = \frac{(\vec{p_3} - \vec{p_1}).(\vec{p_2} - \vec{p_1})}{|\vec{p_2} - \vec{p_1}|^2} \qquad v = \frac{(\vec{p_3} - \vec{p_1}).(\vec{p_2} - \vec{p_1})^{\perp}}{|\vec{p_2} - \vec{p_1}|^2}$$

$$\cos(\alpha) = \frac{(\vec{p_2} - \vec{p_1}).(\vec{p_4} - \vec{p_3})}{|\vec{p_2} - \vec{p_1}||\vec{p_4} - \vec{p_3}|}$$

2.3. Graph Matching

In order to obtain high level semantics, we build a relational graph where each node of this graph corresponds to a segmented part including the feature vector and each arc to their relationship. Matching of the relational graphs of objects with the reference model yields to the detection of objects. The aspect graph of the reference object is formed according to the segmentation results of the training images. The reference graph for real images from a side view is given in Figure 13. We created a reference graph for sketches and three reference graphs for images; namely front-side view, back-side view, and side view images.

The nodes and arcs in the graph have the following attributes.

Unary features:

a) Hu moment invariants; b) compactness; c) eccentricity; d)boundary shape code (turnangle and length of concave axes).

Binary features:

a) Ratio of areas; b) relative position and orientation; c) the adjacency information between nodes with overlapping boundaries or areas.

A system that contain unary and binary classification mappings must also be able to interpret the match and check the conditional rules in order to index the parts correctly. Our solution to this problem is to store the graph representation of the objects.

Although graph matching is widely used for representation of complex objects and scenes [59], [60], [61] and has a long history, it faces problems mostly due to the dependence on the segmentation results. For instance, a graph representation system called Acronym [62] that has been tested on aerial images to classify airplanes, failed when the extracted airplane features were not close enough to expected ones.

To overcome this problem, a new model based segmentation that combines the initial segments or segments them to simpler parts using a feedback from graph representation of the object is proposed. The reference graph representations of the objects are trained from the low level processing results.

Object detection is achieved by matching the relational graphs of objects with the reference model. The number of image segments obtained from the low level segmentation process is S. The input image graph O_n with N nodes ($N \geq S$) and a reference graph (O_r with R nodes) are matched. The aspect graph of the reference object is formed according to the segmentation results of the training images. The attributes of the reference graph nodes are calculated using a training data set. Nodes corresponding to the same object part are extracted to form the reference graph model. The mean μ, variance σ^2 and peak values of the attributes for each node are used in the determination of the thresholds of the matching cost function. After training, in order to detect the OOI in a scene, we form a graph based representation of the segmented regions and their combinations formed by merging them as explained in section 2.1. This input graph is then compared with the reference model.

A reference graph for sketches and three reference graphs for images; namely front-side view, back-side view, and side view images, are created. Our model graphs (for side-view sketches, for three view angles of real cars) have the nodes for mainbody (upper and lower), windows (side, front-side, back-side, front, and back) and tires (front and back).

Given:

- Input image graph O_n with N nodes
- Reference graph O_r with R nodes
- Reference graph node index i
- Input image node index j

Step 1: Match largest reference graph node to the image graph nodes by opening a new branch for every possible match according to the unary descriptors. The total matching cost between node pair $(1, j)$ (matching cost between mainbody from reference graph and all the nodes in the image graph and combination of these nodes) for the branch b is calculated as

$$D^b(1,j) = d_{circ}(1,j) + d_{ecc}(1,j) + d_{mom}(1,j) + d_{curve}(1,j) \tag{7}$$

The differences for the unary descriptors (eccentricity, circularity, moment invariants, and turnangle and length of the concave axes of the boundary) are calculated according to the mean (μ) and deviation (δ) values as given in Eq. (8).

$$d_x(i,j) = \begin{cases} -w_x & \text{if } |x_j - \mu_{x_i}| \leq \delta_{x_i}; \\ w_x & \text{otherwise.} \end{cases} \tag{8}$$

where x_j is the corresponding attribute value of the image node, μ_{x_i} and δ_{x_i} are the mean and deviation values for this node attribute, respectively. These values are obtained from the training data set for sketches and for real images from side, front-side and back-side views. w_x is the weight for the penalty corresponding to this attribute. In our case it is 1.

Step 2: Increase i by 1. For every $j = 1, ..., N$ compute the matching cost $(D^b(i,j))$ between j^{th} and i^{th} node.

The total matching cost for a node pair (i, j) for a branch b $(D^b(i,j))$ is the sum of the unary and binary feature difference as

$$\begin{aligned} D^b(i,j) &= d_{circ}(i,j) + d_{ecc}(i,j) + d_{mom}(i,j) + d_{curve}(i,j) + \\ &\quad d^b_{area}(i,j) + d^b_{adj}(i,j) + d^b_{position}(i,j) \end{aligned} \tag{9}$$

Binary feature differences are computed according to the previously matched nodes for every branch. For every matched node pair (n_i, n_j), the relative area, position, orientation and connectivity are computed between nodes j and n_j and between nodes i and n_i.

The difference of relational area, orientation. and position is calculated using already matched node pairs for the corresponding branch. Let image node n_i be matched to the model node n_j. Corresponding area distance between image node j and model node i for this branch is calculated as

$$d_x^b(i,j) = \begin{cases} -w_x & \text{if } \frac{\mu_{x_i} - \delta_{x_i}}{\mu_{x_{n_i}} + \delta_{x_{n_i}}} \leq \frac{x_j}{x_{n_j}} \leq \frac{\mu_{x_i} + \delta_{x_i}}{\mu_{x_{n_i}} - \delta_{x_{n_i}}}; \\ w_x & \text{otherwise.} \end{cases} \tag{10}$$

where x_j and x_{n_j} are the attribute values for the image nodes i and n_i, and μ and δ are the mean and deviation values, obtained from training, for the corresponding model nodes attribute, respectively. Other attributes are computed similarly.

Step 3: If the matching cost between nodes j and i for a branch is smaller than a threshold found in the training process. set (j, i) as the new matched node pair for this branch. Note that i must be different from the previous n_i's. It is observed that the relative distance of nodes vary highly for different type of cars and especially from sketch to sketch. However, the relative position at a coarser resolution does not change, e.g. the windows are in the upper mainbody (above the concave landmarks) and the tires are below the lower mainbody. The spatial relations (inside or adjacent nodes) are another relational distance. The relative position between the car parts and the mainbody is used as a checking step (Figure 14), e.g. the windows are in the upper mainbody and the tires are below the lower mainbody. The center of gravity of the window must be in the first part of the mainbody which is determined by the highest concavity points and the tires must be adjacent to the concavity points of the second part of the mainbody. Note that this information is not always available as the car can be a hatchback car. or the car can be sketched without these tire concavity points, or there can be occlusion due to other objects or to the view-point.

Step 4: If all the reference graph nodes are taken into account, choose the branch with the maximum number of matched image nodes. If there are more than one resulting branch, choose one with the smallest total matching cost.

Step 5: If majority of reference graph nodes (75%) are matched, decide the presence of OOI, otherwise go to Step 1 for another view class and repeat Steps 1-5 until a match is found for a view class.

3. EXPERIMENTAL RESULTS

The algorithm is implemented on the still images with OOI at the foreground and center of the image, and on video sequences with moving OOI. The object detection is done off-line for the text annotation of images that contain OOI.

The steps of the algorithm are illustrated by using a sketched car example displayed in Figure 15. The segments ($S = 7$) obtained from the initial segmentation process and the corresponding contours are also displayed in this figure. The contour evolution results are displayed in Figure 16 where the resulting landmarks corresponding to high curvature (Eq. 6) values are displayed. The possible combinations ($N = 21$) of these segments to be used in model-based segmentation are displayed in Figure 17. The nodes that are matched to the model graph by using the graph matching algorithm form leaves of the branches that are the candidates for object of interest with semantic segments. The winning branch with the attribute values of the matched nodes is displayed in Figure 18. The matched image nodes are displayed at the left of the figure from the first matched node to the last one. Note that, binary features are computed between each previously matched node and the new node. The unary and binary attribute values for this branch are displayed in Tables 5 and 6, respectively. The values in the square brackets correspond to the attribute range in the model graph (Eq. 8 and 10). The resulting semantic segments that correspond to the car parts are displayed in Figure 19. More sketch images and corresponding segment, node and branch numbers are displayed in Figure 21.

The model graphs for sketches and for real images are obtained by computing the statistics of the node attributes for manually segmented parts. Some training images are displayed in Figure 20. Training data sets for sketches and for real images consist of 40 sketches from the side-view and 70 real images from side, front-side and back-side views. The statistics of some descriptors are displayed in Tables 1, 2, 3, and 4. Table 1 displays the average mean values μ and maximum deviation δ of some attributes (eccentricity, circularity, moment invariant) for the car mainbody obtained from sketches. Table 2, 3, and 4 display the average mean and maximum deviation of some attributes for the car mainbody obtained from the training data sets for different view angles. In Figure 22, some of the real car images from the training data set and the resulting classifications are displayed.

In Figure 23, an example sketch image is displayed. Sixth branch with the maximum number of matched nodes and minimum total matching cost is the result of matching. For this example the number of segments is 5, the number of nodes (number of combinations) is 8, and the number of candidate branches with node pairs having matching cost smaller than the threshold value (5) is 6.

After the determination of the model graph attributes, the algorithm is tested on real images and sketches for several total matching cost values. In Figures 24, 25, 26 and 27 the percentage of correct, false detection and miss versus matching cost threshold are depicted for 18 sketches, 28 side, 23 front-side and 18 back-side images, respectively. In these four figures, the search of an optimum matching cost threshold is displayed. The presented percentage of classification corresponds to the classification results of the image segments. The result of the matching algorithm that gives the image segments and/or their combinations that are matched to car parts is displayed. The number of correct matches divided to the total number of segments of the test images yields the percentage of correct classification of image segments. The falsely classified segments are the segments of the image that are not actually car parts matched to them. The miss percentage is computed from the observation of the algorithm failing to match the image segments that correspond to car parts that are defined in the graph model. As seen in the figures, the false classifications increase in average with increasing threshold while the miss percentage decreases. The optimum threshold for the given algorithm is found to be 5 for sketches and real images. Since the same weighting of dissimilarity is used, it is an expected result to have the same total cost threshold for every subgraph of aspect graph of the real images.

Three examples for region classification for real images with busy and uniform backgrounds are displayed in Figure 28. Note the assumption that the foreground object is about the center of the image eliminates many background regions. However, there are still closed regions adjacent to the OOI. Due to the illumination and color changes the mainbody can be segmented to several regions but the combination of these regions gives the minimum matching cost in graph matching.

In Figure 29, the performance of our method is shown for the Hamburg Taxi video sequence. Initial and final frames, extraction of the OOI from the sequence are shown in the first row. In the second row, the segmentation and matching results are displayed. In hierarchical object description, each segment helps us in handling the problems caused by occlusion. For instance, the tires of the moving car in the Hamburg Taxi video sequence are not visible, however, the number of matched nodes are high enough to detect the car. Therefore, the algorithm works for partial occlusion of the object.

The most similar work to our graph-based object detection scheme is the object based retrieval system proposed by Xu et al [28]. The multi-level segmentation scheme used to create semantic features is similar to our model-based segmentation scheme. Our work mainly differentiates from the authors' algorithm at the bottom level of the segmentation tree. The authors form the root of the tree by grouping pixels similar in color therefore restricting the concept of homogeneity to color. Our approach is based on the observation that although complex objects can have shape and color variability within subparts of different objects, the relation between the subparts and the primitive shape characteristics are highly preserved. Therefore, our homogeneity concept is based on color and curvature, and the lowest level is formed of simplest visual parts in terms of curvature and color. Hence, the shape attributes are chosen so that the fundamental shape characteristics are captured as opposed to B-spline fit of the complex region boundaries as described in [28]. The authors match a given query template (e.g. car mainbody) to database images in a top-down fashion where the relation of other subparts is not used. However. although separating regions to primitive parts and combining them according to model-based segmentation increases the computational complexity, it enables to use the relations of basic subparts for a more robust detection scheme in terms of high-variability within class and occlusion. Furthermore, it is shown that the same graph-based object representation is suitable for non-rigid object detection i.e., human bodies with different postures since the lowest level of the tree can capture the articulated movements [68].

4. CONCLUSIONS AND FUTURE WORK

In this paper, we present an object-based image and video retrieval algorithm for car detection purposes. The work focuses on the problem of connecting low level features to high level semantics by developing relational object presentation.

The paper first examines extraction of low level features from images and videos using intensity, color and motion of pixels and regions. Local consistency based on these features and geometrical characteristics of the regions is used to group object parts. The problem of managing the segmentation process is solved by a new approach that uses object based knowledge in order to group the regions according to a global consistency. A new model-based segmentation algorithm is introduced that uses a feedback from relational representation of the object. Object detection is achieved by matching the relational graphs of objects with the reference model. The algorithm maps the attributes, interprets the match and checks the conditional rules in order to index the parts correctly. The major advantages can be summarized as improving the object extraction by reducing the dependence on the low level segmentation process and combining the boundary and region properties. Furthermore, the features used for segmentation are also attributes for object detection in relational graph representation. This property enables to adapt the segmentation thresholds by a model-based training system. The detection rate corresponds to correct classification of object parts. The detection rate is 83% for free-hand car sketches and 87%, 76% and 77% for real car images viewed from side, front-side and back-side respectively. The test data set includes images and sequences from different sources and at different resolutions and occlusions. Object detection for automatic image and video annotation must deal with high-variability within object class, image resolution and different object classes. The detection scheme presented in this paper is scalable in terms of variety of object appearance since the node model structure and attributes range are flexible for the detection of object types from low to high within-class variability. The relational object representation with model based segmentation is scalable for different resolution levels; from detecting objects with a few number of object parts to detecting objects with many parts. Furthermore, the scheme can be trained for new object types and its scalability is shown in our current work for human detection and activity recognition. Our current and future work include object detection and activity recognition in uncompressed and compressed images (JPEG) and videos (MPEG) [69] where the graphical representation is used for non-rigid objects.

REFERENCES

1. I. B. Ozer, W. Wolf, and A. N. Akansu, "A Graph Based Object Description for Information Retrieval in Digital image and Video Libraries", CBAIVL, pp.79-83, June 1999.

2. R. Jain, "Workshop Report: NSF Workshop on Visual Information Management Systems", *Proc. SPIE Conf. on Vis. Commun. and Image Proc.*, 1993.

3. R. Jain, A. Pentland, and D. Petkovic, "NSF-ARPA Workshop on Visual Information Management Systems", Cambridge, MA, June 1995.

4. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafine, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System", *IEEE Computer*, 1995.

5. J. Dowe, "Content-based Retrieval in Multimedia Imaging", *Proc. SPIE Conf. on Vis. Commun. and Image Proc.*, 1993.

6. B. Furht, S.W. Smoliar, and H. Zang, "Video and Image Processing in Multimedia System", Kluwer Academic Publishers, 1995.

7. R.W. Picard and T.P. Minka, "Vision Texture for Annotation", MIT Multimedia Laboratory Perceptual Computing Section TR No.302, 1995.

8. S. Scraloff and A. Pentland, "Modal Matching for Correspondence and Recognition", *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 17, pp.545-561, 1995.

9. J.R. Smith and S.F. Chang, "VisualSEEk: A Fully Automated Content-Based Image Query System", *Proc. ACM Multimedia Conf.*, pp. 87-98, Boston, 1996.

10. S.F. Chang, W. Chen, and H. Sundaram, "Semantic Visual Templates - Linking Visual Features to Semantics", *Proc. Int. Conf. on Image Proc.*, 1998.

11. H. Yu, W. Wolf, "A Visual Search System for Video and Image Databases", *Proc. IEEE Multimedia*, 1997.

12. J. Zhang, H.Krim, and X. Zhang, "Invariant Object Recognition by Shape Space Analysis", *Proc. Int. Conf. on Image Proc.*, 1998.

13. T.S. Huang, S. Mehrotra, and K. Ramchandran, "Multimedia Analysis and Retrieval System(MARS) Project", *Proc. of 33rd Annual Clinic on Library Application of Data Processing-Digital Image Access and Retrieval*, 1996.

14. A. Yoshitaka, T. Ichikawa, "A survey on Content-Based Retrieval for Multimedia Databases", IEEE Trans. on Knowledge and Data Eng., Vol 11, No 1, pp. 81-92, Jan/Feb. 1999.

15. A. Gupta, R. Jain, "Visual Information Retrieval", Communications of ACM, Vol. 40, No 5, pp 70-79, May 1997.

16. A. H. Barr, "Superquadrics and Angle Preserving Deformations," IEEE Computer Graphics Applications, vol. 1, pp. 11-23, 1981.

17. Y. Rui, T. S. Huang, and S. Chang, "Image Retrieval: Current Techniques, Promising Directions and Open Issues" , Journal of Visual Communication and Image Representation , Vol. 10, 39-62, March, 1999.

18. A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, "Content-based image retrieval at the end of the early years", IEEE Transcations on Pattern Analysis and Machine Intelligence, Volume: 22, Issue: 12, pp. 1349-1380, Dec. 2000.

19. F. Solina and R. Bajcsy, "Recovery of parametric models from range images: the case for superquadrics with global deformations," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 12, no. 2, pp. 131-147, Feb. 1990.

20. A. Pentland, R. Picard, and S. Sclaroff "Photobook: Tools for Content Based Manupulation of Image Databases", Storage and Retrieval of Image and Video Databases II, Paper No. 2185-05, San Jose, Calif., pp.34-47, SPIE, Feb. 1994.

21. A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based Manipulation of Image Databases", International Journal of Computer Vision, 1996.

22. V.E. Ogle and M. Stonebraker, "Chabot: Retrieval from a Relational Database of Images", *Computer*, vol. 28, no.9, Sept. 1995.

23. J. R. Smith and S. F. Chang, "Visually Searching the Web for Content", IEEE Multimedia, Vol 4, No 3, pp 12-20, July/Sept 1997.

24. W.-S. Li and K.S. Candan, "SEMCOG: A Hybrid Object-based Image Database System and Its Modeling, Language, and Query Processing", Proceedings of the 14th International Conference on Data Engineering, pp. 284-291, Feb. 1998.

25. M.P. Dubuisson, S. Lackshmanan, and A.K. Jain, "Vechicle Segmentation and Classification Using Deformable Templates", *IEEE Trans. Pattern Analysis Mach. Intell.* , pp. 293-307, March 1996.

26. M.P. Dubuisson, A.K. Jain, and W.C. Taylor, "Segmentation and Matching of Vehicles in Road Images", *Transportation Research Report*, No.1412, pp.57-63.

27. A. Jain, Y. Zhong, and S. Lakshmanan, "Object Matching Using Deformable Templates", *IEEE Trans. Pattern Analysis Mach. Intell.* , pp. 408-439, March 1996.

28. Y. Xu, E. Saber, and A.M. Tekalp, "Object Formation by Learning in Visual Databases Using Hierarchical Content Description", *Proc. Int. Conf. on Image Proc.*, October 1999.

29. S. Loncaric, "A Survey of Shape Analysis Techniques", Pattern Recognition, Vol 31, No 8, pp 983-1001, 1998.

30. R.N. Haber, M. Hershenson, The Psychology of Visual Perception, Holt, Rinehart and Winston Inc, 1973.

31. L. Zusne, "Visual Perception of Form", Academic Press, B200, New York, 1970.

32. D.D. Hoffman and W.A. Richards, "Parts of Recognition", Cognition, vol. 18, pp. 65-96, 1984.

33. D.O. Hebb, "The Organization of Behaviour", Wiley, New York, 1949.

34. S.Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, California, pp. 232-237, June 1998.

35. J.R. Bennet and J.S. McDonald, "On the Measurement of Curvature in a Quantized Environment", IEEE Trans. on Comput., vol. 24, pp. 803-820, 1975.

36. E. M Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem and J. S. B. Mitchell, "An efficiently computable metric for comparing polygonal shapes", IEEE Trans. Pattern Anal. Mach. Int, Vol 13, pp 209-216, 1986.

37. C. P. Papageorgiou, T. Poggio, "A Trainable Object Detection System: Car Detection in Static Images", Technical paper, MIT, CBCL Paper no. 180, Oct. 1999.

38. C. P. Papageorgiou, M. Oren and T. Poggio, "Pedestrian Detection Using Wavelet Templates," Proc. of CVPR, pp. 193-199, June 1997.

39. L. J. Latecki and A. Rosenfeld, "Supportedness and tameness: Differentialless geometry of plane curves", Pattern Recognition, Vol 31, pp 607-622, 1998.

40. C.C. Chang, S.M. Hwang, and D.J. Buehrer "A Shape Recognition Scheme Based on Relative Distances of Feature Points from the Centroid", Pattern Recognition, vol. 24, pp. 1053-1063, 1991.

41. H. Freeman, "On the Encoding of Arbitrary Geometric Configurations", IRE Transactions, vol. 10, pp. 260-268, 1961.

42. E. Persoon and K.S. Fu, "Shape Discrimination Using Fourier Descriptors", *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 8, pp.388-397, 1986.

43. S. Wang, P. Chen, and W. Lin, "Invariant Pattern Recognition by Moment Fourier Descriptor", Pattern Recognition., vol. 27, pp. 1735-1742, 1994.

44. A. Bengtsson and J. Eklundh, "Shape Representation by Multiscale Contour Approximation", IEEE PAMI, vol. 13, pp. 85-93, 1991.

45. F.S. Cohen, Z. Huang, and Z. Yang, " Invariant Matching and Identification of Curves Using B-splines Curve Representation", *IEEE Trans. on Image Processing*, vol. 4, pp.1-10, 1995.

46. B. Gunsel and A.M. Tekalp, "Shape Similarity Matching for Query by Example", *Pattern Recognition*, vol. 31, No. 7, pp.931-944, July 1998.

47. A. P. Witkin, "Scale Space Filtering", Proc. 8th Int. Joint Conf. on Artificial Intelligence, pp. 1019-1022, 1983.

48. H. Asada and M. Brady, "The Curvature Primal Sketch", IEEE PAMI, vol. 8, pp. 2-14, 1986.

49. F. Mokhtarian and A.K. Mackworth, "A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves", IEEE PAMI, vol. 14, pp. 789-805, 1992.

50. L.J. Latecki and R. Lakamper, "Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution", *Computer Vision and Image Understanding*, vol. 73, no.3, pp.441-454, 1999.

51. J.L. Mundy and A. Zisserman, "Geometric Invariance in Computer Vision", MIT Press, 1992.

52. M.K. Hu, "Visual Pattern Recognition by Moment Invariants", IRE Trans. Inform. Theory, vol. 8, pp. 179-187, 1962.

53. H. Blum and R. Nagel, "Shape Description Using Weighted Symmetric Axis Features", Pattern Recognition, vol. 10, pp. 167-180, 1978.

54. K. Siddiqi, A. Shokoufandeh, S. J. Dickinson and W. Zucker, "Shock Graphs and Shape Matching", Technical Report, http://www.cim.mcgill.ca/ siddiqi/journal.html.

55. F. Leymarie and M.D. Levine, "Simulating the Grassfire Transform Using an Active Contour Model", PAMI, vol. 14, pp. 56-75, 1992.

56. J. Shi and C. Tomasi, "Good Features to Track", *CVPR*, 1994.

57. K. Harris, S.N. Efstratiadis, N. Maglaveras, and A.K. Katsaggelos, "Hybrid Image Segmentation Using Water Sheds and Fast Region Merging", *IEEE Trans. on Image Processing*, vol. 7, pp.1684-1699, 1998.

58. J. B. Burns, R. S. Weiss and E. M. Riseman, "View Variation of Point-Set and Line-Segment Features", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, No 1, pp. 51-68, 1993

59. D.H. Ballard and C.M. Brown, "Computer Vision", Prentice-Hall, Englewood Cliffs, NJ, 1982.

60. T. Caelli and W.F. Bischof, "Machine Learning and Image Interpretation", Plenum Press, New York, NY, 1997.

61. R.M. Haralick and L.G. Shapiro, "Computer and Robot Vision", Addison Wesley Publishing Co., 1993.

62. R. A. Brooks, "Model-Based Three Dimensional Interpretations of Two Dimensional Images", PAMI, vol. 5, pp. 140-150, 1983.

63. M. Nagao, T. Matsuyama, Y. Ikeda,"Region Extraction and Shape Analysis in Aerial Photographs", CGIP, pp. 195-223, 1979

64. M. Kass, A. P. Witkin and D. Terzopoulos. "Snakes: Active contour models", Inter. Journal on Comp. Vision, Vol 1, No 4. pp. 321-331, 1988

65. H. H. S. Ip and D. Shen,"An Affine Invariant Active Contour Model for Model-Based Segmentation", IVC, pp. 135-146, 1998

66. L. Liu, S. Sclaroff,"Deformable Shape Detection and Description via Model-Based Region Grouping", CVPR 1999

67. F. Mokhtarian and A. Mackworth, "Scale Based Description and Recognition of Planar Curves and 2D Shapes", *IEEE PAMI*, vol. 8(1), pp. 34-43, 1986.

68. I. B. Ozer, W. Wolf, A. N. Akansu,"Relational Graph Matching for Human Detection and Posture Recognition", SPIE, Photonic East 2000, Internet Multimedia Management Systems, Boston, November 2000.

69. I. B. Ozer, W. Wolf, A. N. Akansu, "Human activity detection in MPEG sequences", IEEE Workshop on Human Motion, pp. 61-66, 2000.

TABLE 1

Mean and maximum deviation of the unary descriptors for the sketched car mainbody.

	Mean	Maximum deviation
Eccentricity	15.93	12.19
Circularity	0.39	0.11
First Moment Invariant	0.44	0.12

TABLE 2

Mean and maximum deviation of the descriptors for the mainbody of the real car image from side view.

	Mean	Maximum deviation
Eccentricity	20.30	9.01
Circularity	0.25	0.15
First Moment Invariant	0.50	0.094

TABLE 3

Mean and maximum deviation of the descriptors for the mainbody of the real car image from front-side view.

	Mean	Maximum deviation
Eccentricity	13.21	9.99
Circularity	0.1524	0.0961
First Moment Invariant	0.4618	0.1677

TABLE 4

Mean and maximum deviation of the descriptors for the mainbody of the real car image from back-side view.

	Mean	Maximum deviation
Eccentricity	13.0515	7.6203
Circularity	0.175	0.0577
First Moment Invariant	0.4353	0.1644

TABLE 5

Unary features (U) for the example given in Figure 18: The values in the square brackets are the mean (μ_{x_i}) and maximum deviation (δ_{x_i}) values obtained from training data set. The values before the square brackets are the unary attribute values (x_j) of the corresponding image node.

	Eccentricity	Circularity	First Moment Invariant	Turnangle	Normalized Lenght of the concave axes
U1	8.33 [15.93,12.19]	0.46 [0.39,0.11]	0.31 [0.44,0.12]	14.137 [12.3,17.7]	0.28 [0.2,0.47]
U2	1.24 [1.5,0.8]	1.08 [1.17,0.15]	0.16 [0.16,0.02]	15.71 [12.4,14.8]	0 [0,0.01]
U3	1.12 [1.5,0.8]	1.27 [1.17,0.15]	0.16 [0.16,0.02]	12.57 [12.4,14.8]	0 [0,0.01]
U4	4.84 [3.5,1.5]	0.89 [0.92,0.2]	0.21 [0.20,0.02]	9.43 [8.7,12.9]	0 [0,0.01]
U5	3.15 [3.5,1.5]	0.97 [0.92,0.2]	0.19 [0.20,0.02]	9.42 [8.7,12.9]	0 [0,0.01]

TABLE 6
Binary features (B) for the example given in Figure 18: The values in the square brackets are the binary values $(B(\mu_{x_i}, \mu_{x_{n_i}}, \delta_{x_i}, \delta_{x_{n_i}}))$ obtained from training data set. Note that the relative orientation has two intervals since in general the major and minor axes of sketched windows and tires are comparable. The values before the square brackets are the binary attribute values $(B(x_j, x_{n_j})$) of the corresponding image node.

	Area ratio	Relative position	Relative orientation
B2-1	7.25 [7.46,15.92]	3.44 [2.75,7.60]	0.061 [0/0.9,0.2/1.0]
B3-1	9.83 [7.46,15.92]	6.77 [2.75, 7.60]	0.99 [0/0.9,0.2/1.0]
B3-2	1.35 [0.70,1.3]	10.00 [6.7,9.3]	0.22 [0/0.9,0.2/1.0]
B4-1	8.88 [6.80,10.2]	1.45 [1.31,2.07]	1.00 [0/0.9,0.2/1.0]
B4-2	1.22 [0.63,1.5]	2.70 [3.4,6.87]	0.063 [0/0.9,0.2/1.0]
B4-3	0.90 [0.63,1.5]	4.85 [3.4,6.87]	0.99 [0/0.9,0.2/1.0]
B5-1	11.66 [6.80,10.2]	1.98 [1.31,2.07]	0.99 [0/0.9,0.2/1.0]
B5-2	1.61 [0.63,1.5]	6.10 [3.4,6.87]	0.17 [0/0.9,0.2/1.0]
B5-3	1.19 [0.63,1.5]	5.25 [3.4,6.87]	0.99 [0/0.9,0.2/1.0]
B5-4	1.31 [0.6,1.7]	3.2 [1.8,3.78]	0.99 [0/0.9,0.2/1.0]

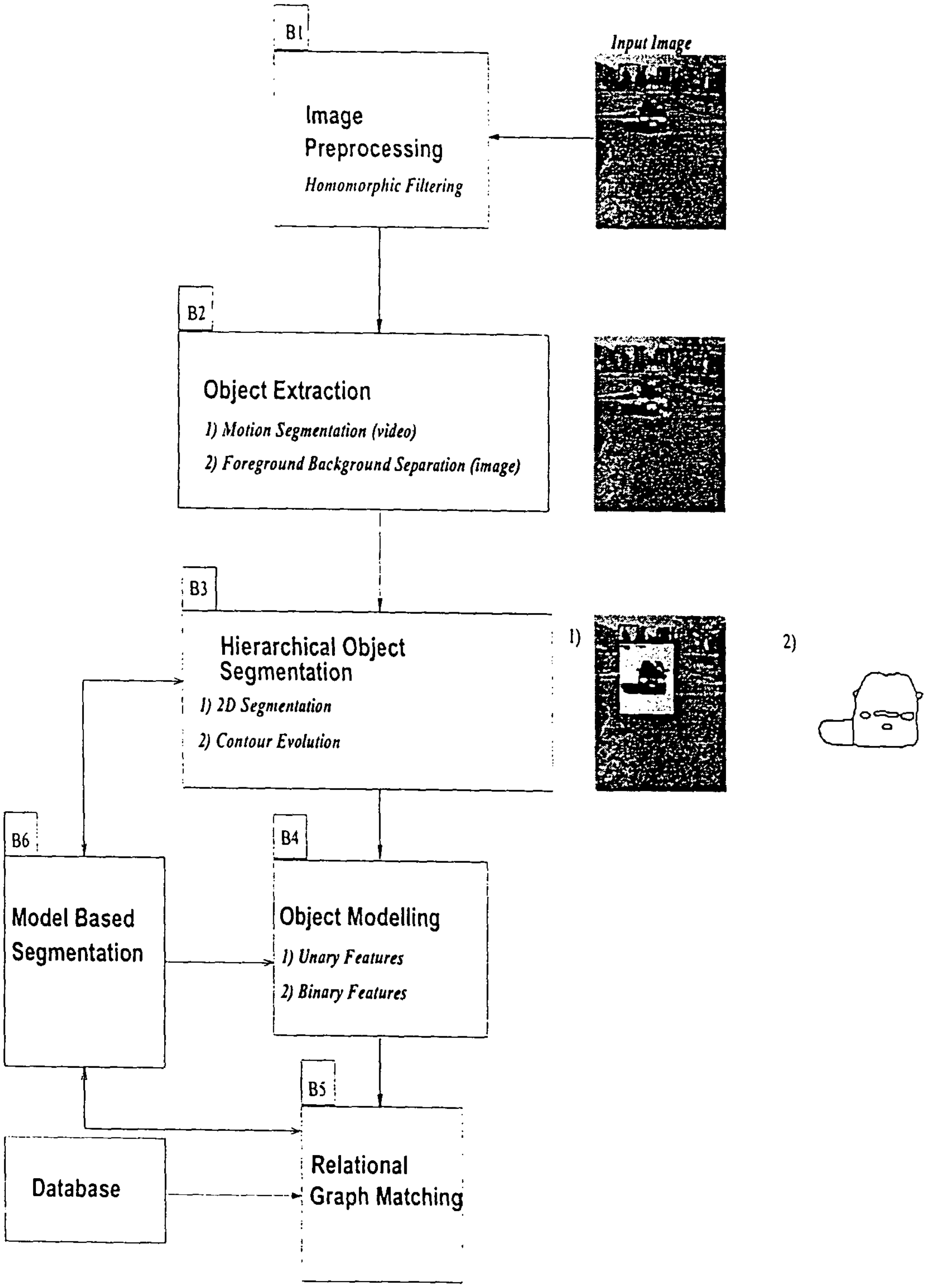


FIG. 1. Block diagram of the proposed retrieval system.

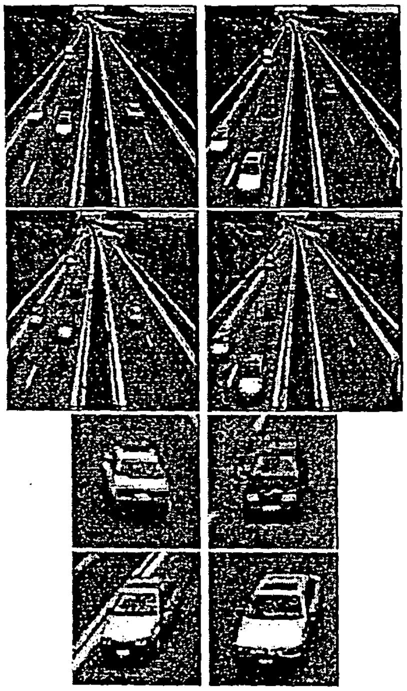


FIG. 2. Extraction of moving objects in a MPEG-7 video sequence. First Row: Initial and final video frames; Middle Row: Tracked features (motion threshold = 1pixel/frame, distance threshold = 15 pixels); Bottom Rows: Potential areas that contain OOI.

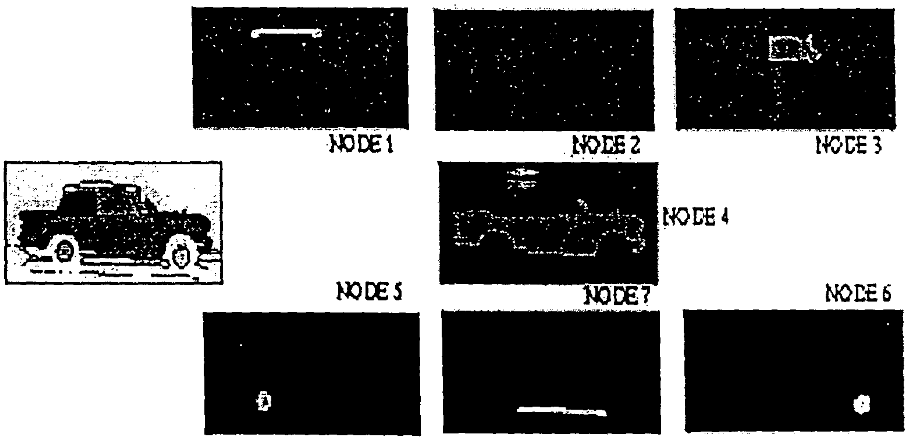


FIG. 3. Example car image and corresponding segments.

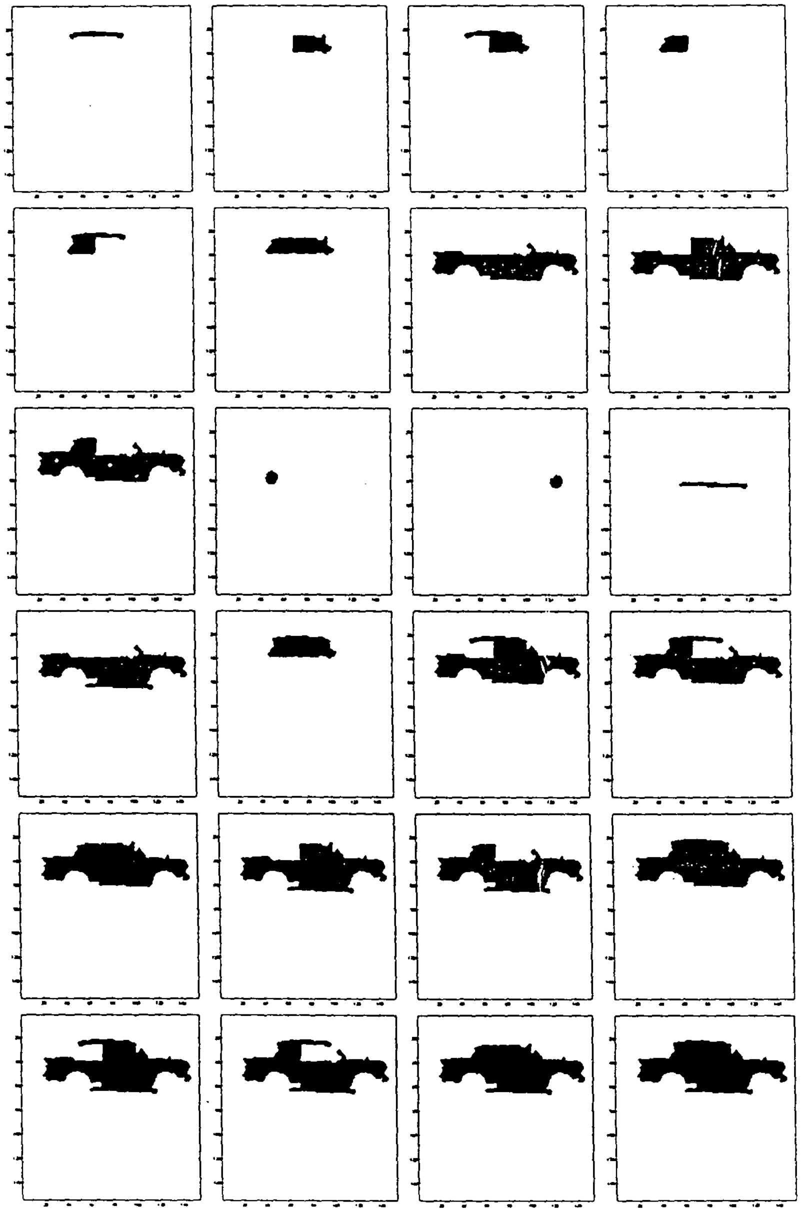


FIG. 4. Input image nodes (N=24, S=7)

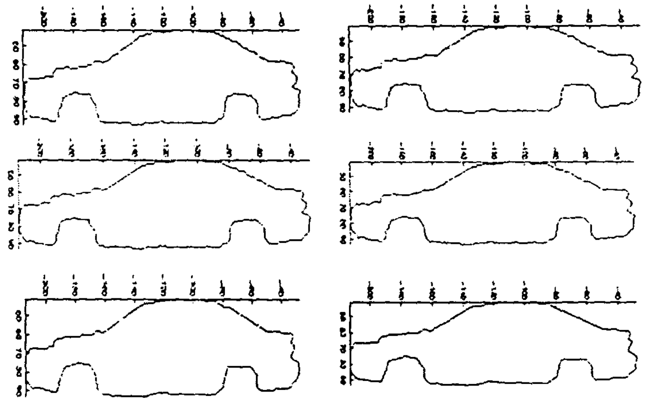


FIG. 8. Selected stages of the discrete curve evolution for the car mainbody.

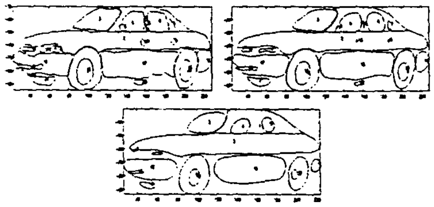


FIG. 9. Multiscale representation of car segments. Top left: σ of the Gaussian kernel = 1.5, Top right: $\sigma = 5$, Bottom: $\sigma = 10$.

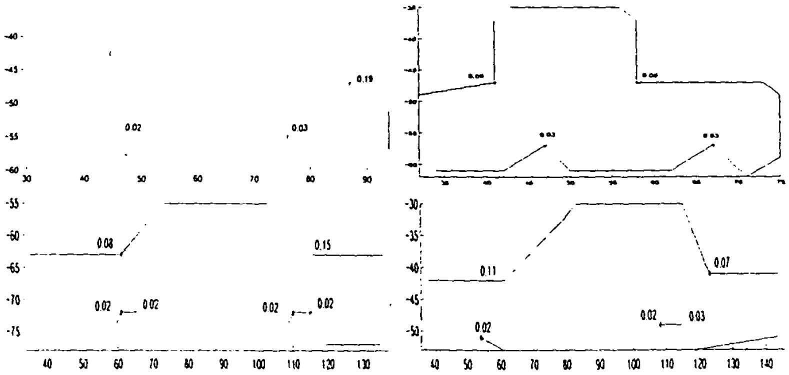


FIG. 10. Concave points of mainbody for sketches.

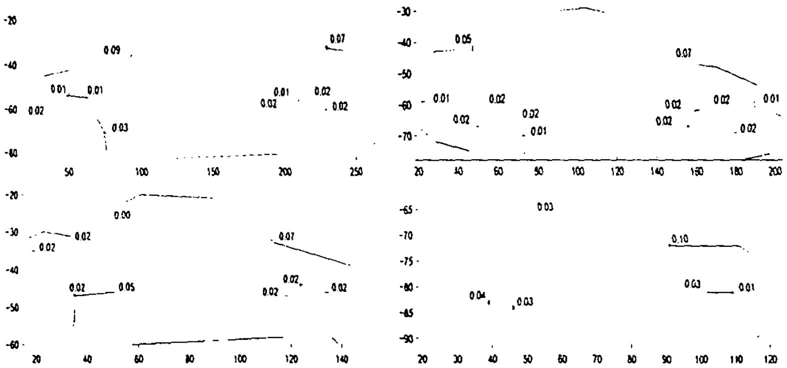


FIG. 11. Concave points of mainbody for real images from side view.

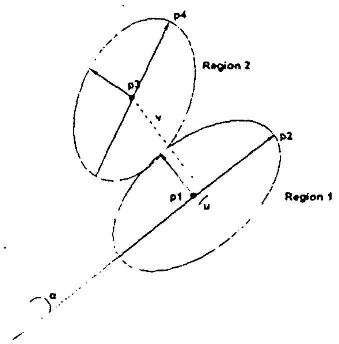


FIG. 12. Relative position and orientation of two regions.

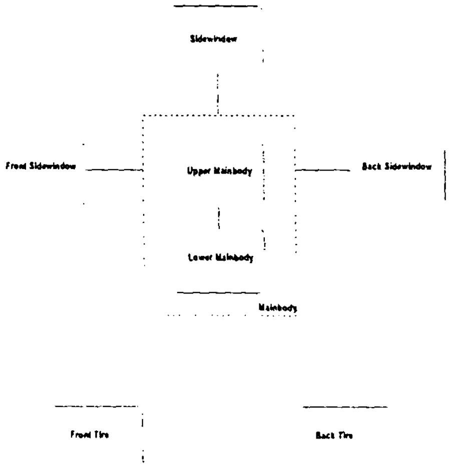


FIG. 13. Reference graph from side view for real car images

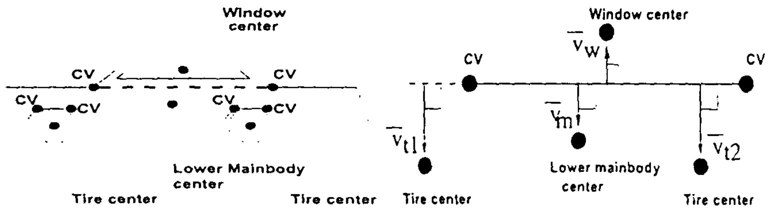


FIG. 14. Left: Possible concavity and center points of a side car, Right: Normal vectors of the line between the maximum concavity points.

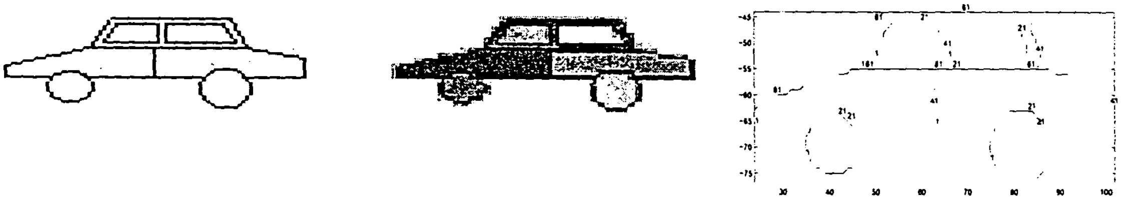


FIG. 15. An example for classification of parts of a sketched car. Left: Original image; Middle: Segments obtained from initial segmentation process; Right: Segment Contours

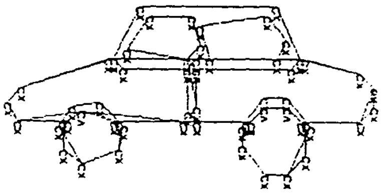


FIG. 16. Curve evolution result with marked landmarks.

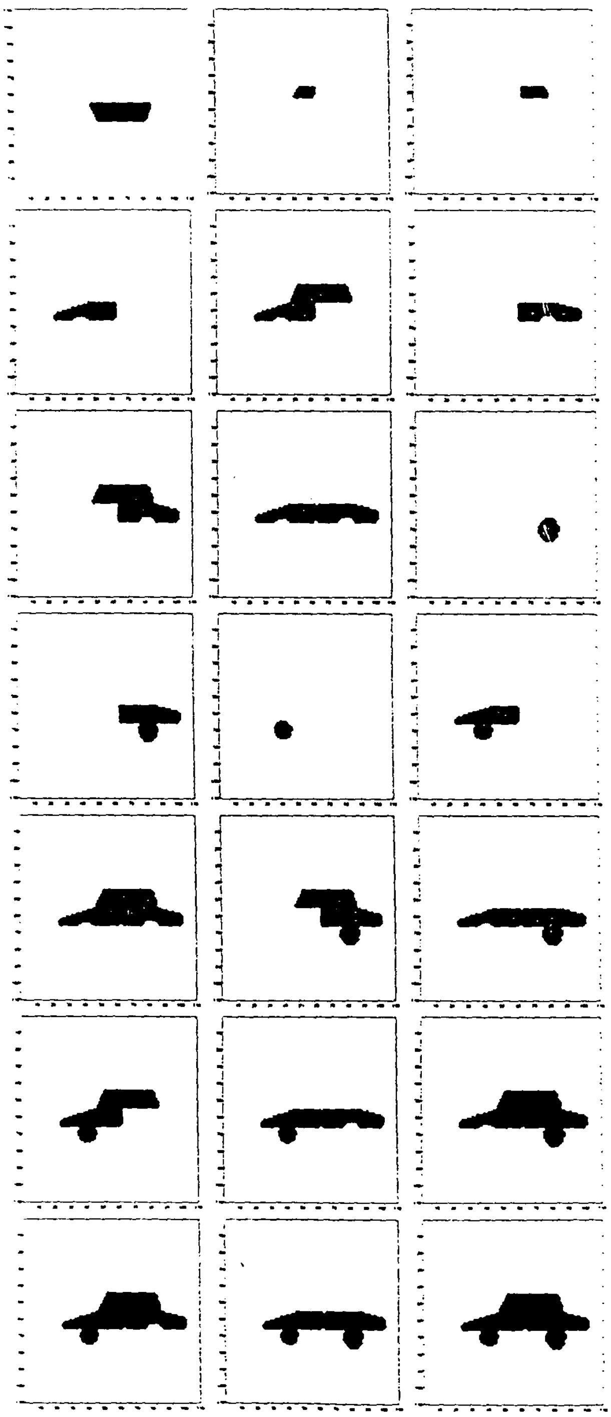


FIG. 17. Input image nodes (N=21, S=7)

78 OF 141

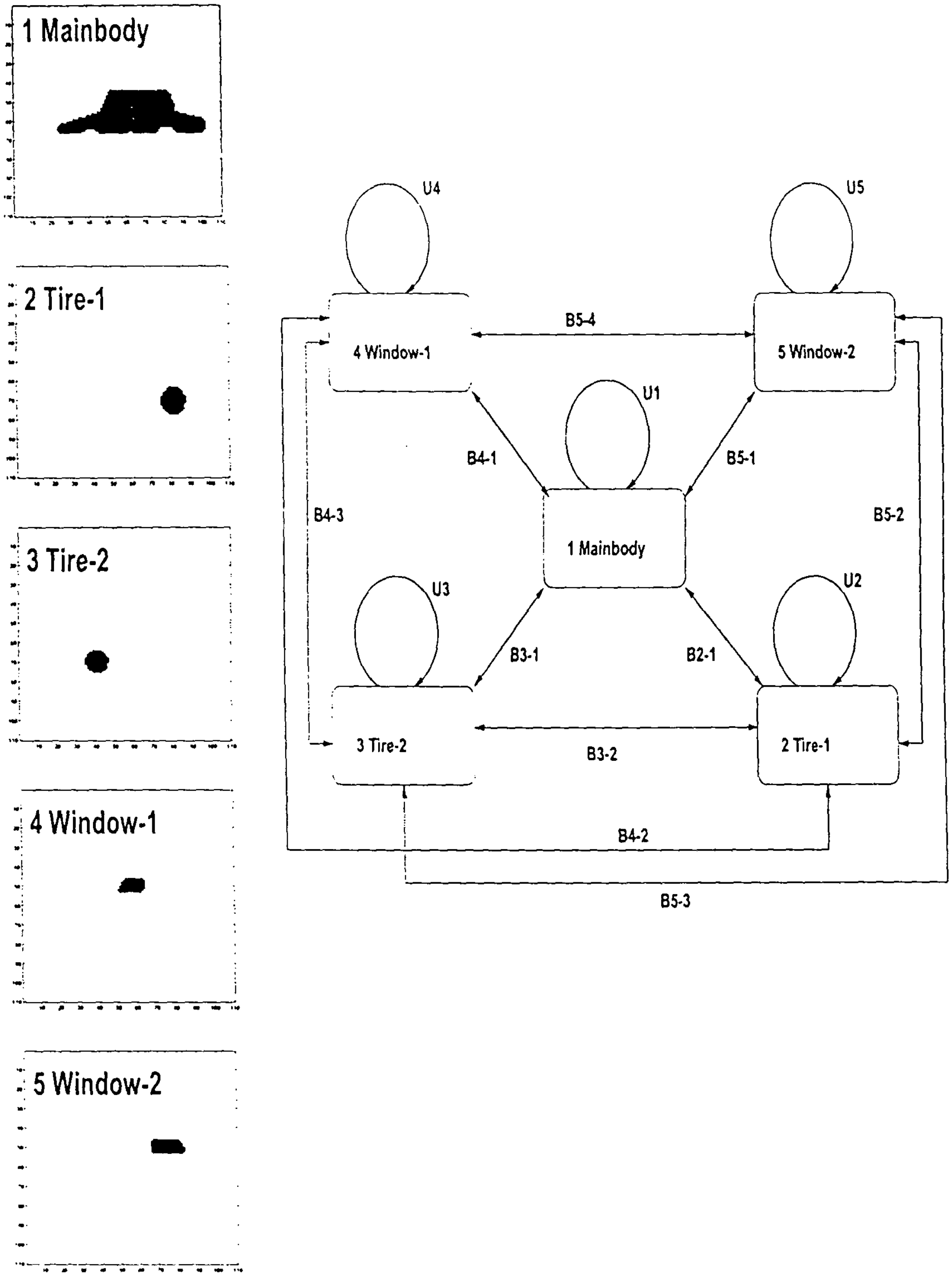


FIG. 18. Graph matching result for the example displayed in Figure 15. Uj's denote the unary attributes for matched nodes. $Bj-n_j$'s denote the binary attributes between the j^{th} node and n_j^{th} previously matched node of the branch.

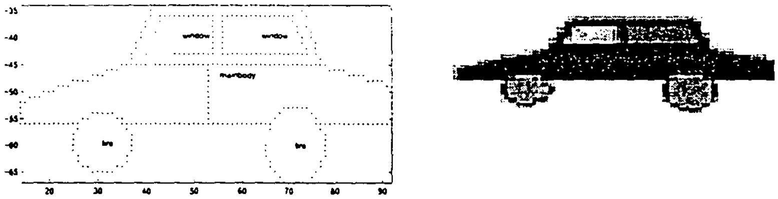


FIG. 19. Left: Classification of nodes after graph matching; Right: Semantic segments that correspond to mainbody, windows and tires of the car.

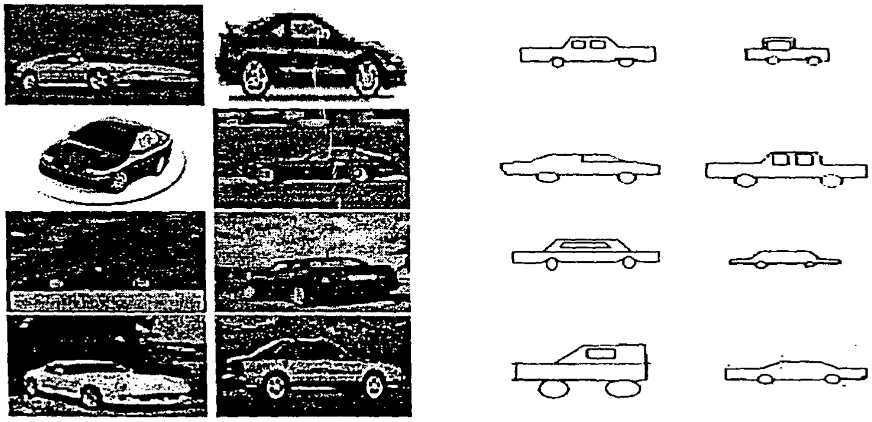


FIG. 20. Left two columns: Training set images, Right two columns: Training set images obtained from student sketches

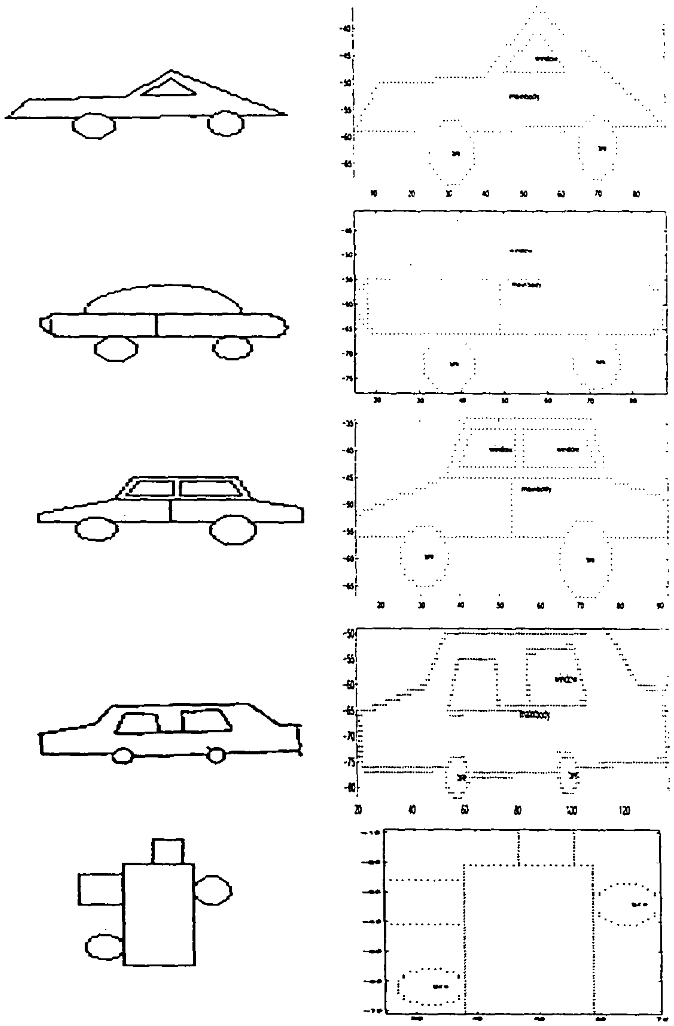


FIG. 21. First row: Original sketch and matching result (all segments are correctly classified), number of segments(n_s)= 4, number of nodes(n_n)= 7, number of candidate branches(n_b)= 5, Second row: All segments are correctly classified, mainbody is the combination of three segments, n_s=6, n_n=19, n_b=19, Third row: All segments are correctly classified, mainbody is the combination of three segments, n_s=7, n_n=21, n_b=321, Fourth row: One window is missing, because the center of the gravity of the window is below the concavity line. n_s=5, n_n=8, n_b=5, Fifth row: For the non-car sketch the matched parts are not correct, the total segment number is 5 and the number of matched segments is 2 which is not sufficient to decide for the presence of OOI, n_s=5, n_n=20, n_b=48.

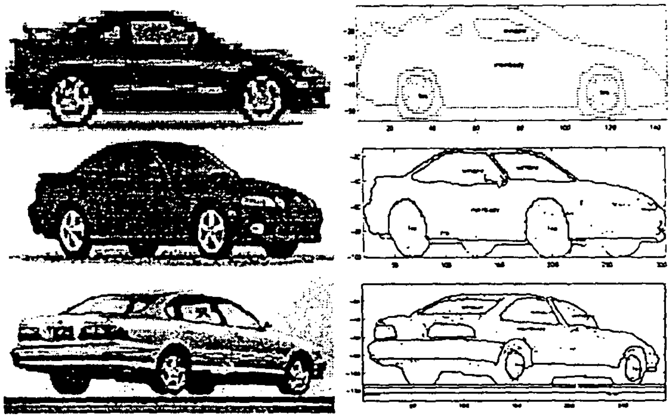
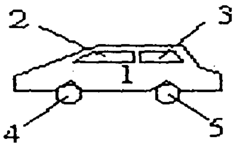


FIG. 22. Training examples: Left: Original image, Right: Classification result



Branch number:	1	2	3
	Mainbody 2, dif=3	Mainbody 3, dif=4	Mainbody 4 1, dif=2
	Total dif = 3	Total dif = 4	Total dif = 2

	4	5	6
	Mainbody 5 1, dif=2 Tire 4, dif=-2 Window 2, dif=-2 Window 3, dif=-3	Mainbody 5 1 4, dif=3 Window 5, dif=-1	Mainbody 1, dif=2 Tire 5, dif=-3 Tire 4, dif=-3 Window 2, dif=-2 Window 3, dif=-3
	Total dif = -4	Total dif = -4	Total dif = -9

FIG. 23. Top: Original sketch with segment numbers, Bottom: Resulting branches, where the sixth branch has the maximum number of matched nodes and minimum total difference. The segment numbers 6, 7 and 8 are the combinations of segments 4 and 1; 5 and 1; 5, 1 and 4 respectively.

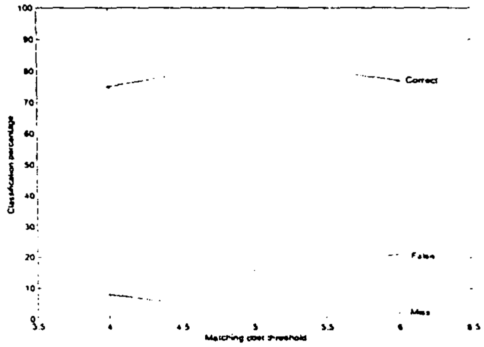


FIG. 24. Classification percentage of correct, false detection and miss for nodes versus matching cost threshold for 18 sketches.

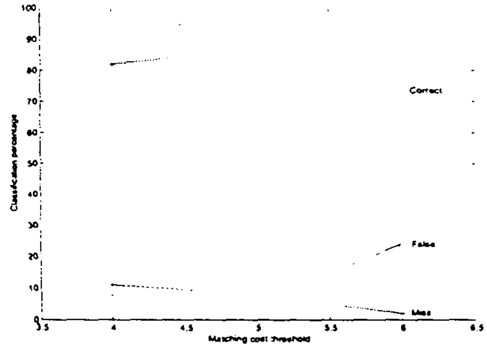


FIG. 25. Classification percentage of correct, false detection and miss for nodes versus matching cost threshold for 28 real side images.

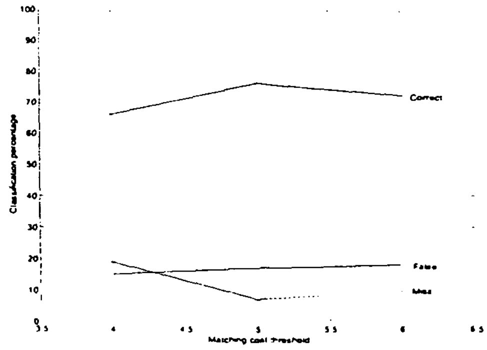


FIG. 26. Classification percentage of correct, false detection and miss for nodes versus matching cost threshold for 23 real front-side images.

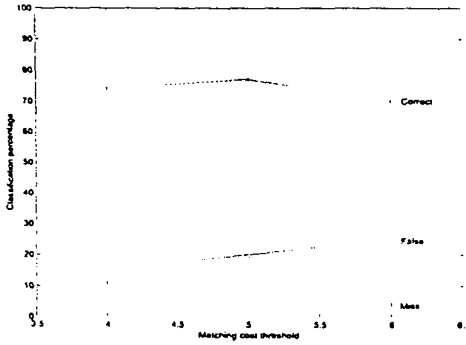


FIG. 27. Classification percentage of correct, false detection and miss for nodes versus matching cost threshold for 18 real back-side images.

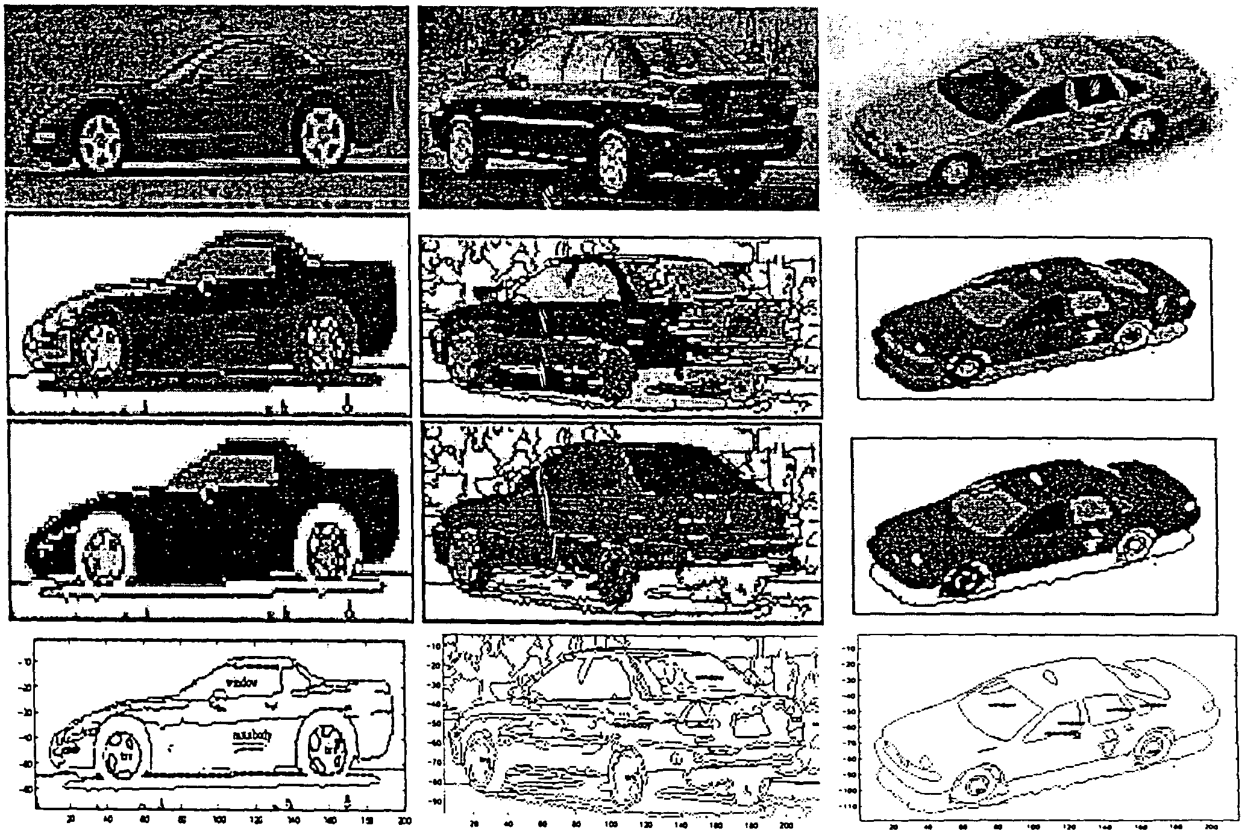


FIG. 28. First row: Original images with busy and uniform backgrounds; Second row: Segmented images after taking only the closed regions at the center part of the image and after eliminating/merging small regions; Third row: Classification of nodes after graph matching; Fourth row: Resulting nodes.

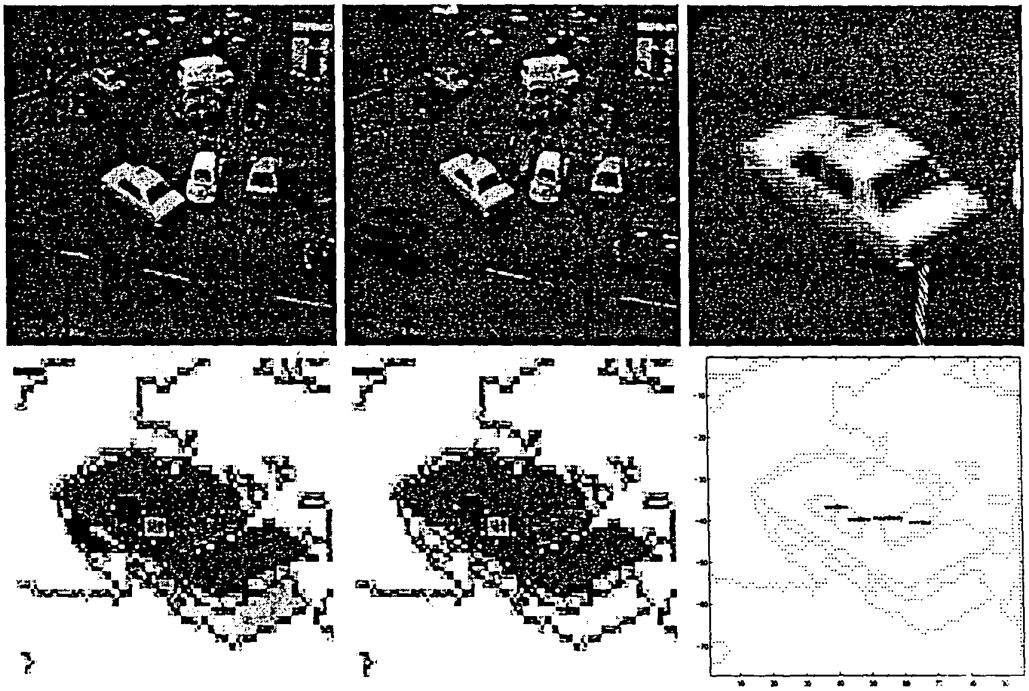


FIG. 29. Separation of a moving car in Hamburg Taxi video sequence and matching result.

HUMAN DETECTION IN COMPRESSED DOMAIN

I. Burak Ozer and Wayne Wolf

Department of Electrical Engineering,
Princeton University,
Princeton, NJ 08540, USA.
{iozer,wolf}@ee.princeton.edu

ABSTRACT

In this paper, we propose an algorithm for human detection in JPEG compressed still images and MPEG I frames. In this new algorithm, the overall shape of a standing or walking person is detected by using an eigenspace representation of human silhouettes obtained from AC-DCT coefficients. Our approach is invariant to changes in intensity, color and textures and has the advantage of using the available data in the standard compression algorithms. The algorithm achieves a correct detection rate of 80% for frontal and rear views of human body in cluttered scenes.

1. INTRODUCTION

Most human activity recognition and detection techniques are done in the uncompressed domain and depend on proper segmentation of the human body. However, for large libraries, compressed domain image/video processing for existing compression standards can solve the problem of bandwidth and intensive computing. The purpose of this work is human detection in still images and video frames in the compressed domain in order to reduce computational complexity and avoid dependence on correct segmentation in an uncompressed image.

Most of the retrieval systems that are based on the compression schemes are devised for particular objects. Photobook [1] project uses a compact eigenspace representation of faces that can be used for both recognition as well as image compression. In [2], the structural information of pedestrians is presented by a subset of wavelet coefficients and pedestrians are detected by the support vector machine classification method. Our work aims to retrieve information from images and videos compressed using standard algorithms such as JPEG and MPEG. This differentiates our approach from the previous work where the compression algorithms are governed by characteristics of object of interest to be retrieved. In our algorithm, the overall shape of a standing or walking person (from front or back-view) in still images is detected by using the AC-DCT coefficients.

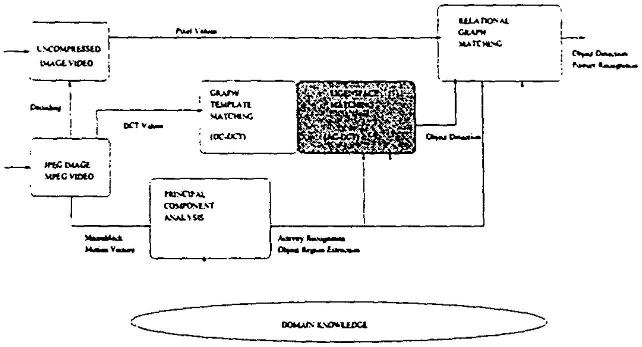


Figure 1: Object detection and activity recognition system with the human detection by eigenspace matching for compressed domain images/video frames (shaded region).

The use of available information in compressed video and images has been investigated mostly for video indexing, and shot and scene classification. The object detection in the compressed domain is more restricted since this application requires more detailed information. Schonfeld [3] proposes an object tracking algorithm by using compressed video only with periodically decoding I-frames. The object to be tracked is initially detected by an accurate but computationally expensive object detector applied to decoded I-frames. Zhong et al. [4] automatically localize captions in JPEG compressed images and I frames of MPEG compressed videos. Intensity variation information encoded in the DCT domain is used to capture the directionality and periodicity of blocks. Wang [5] proposes an algorithm to detect human face regions from dequantized DCT coefficients of MPEG video. This method is suitable for color images with face regions greater than 48 by 48 pixels (3 by 3 MPEG macroblocks).

860F 141

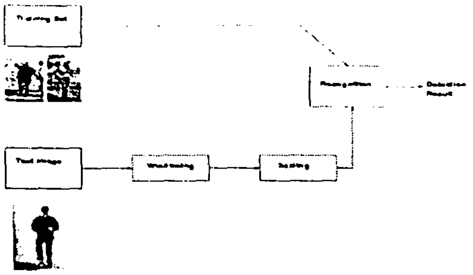


Figure 2: Human detection system for low resolution JPEG images.

However, usually, the skin information from the DCT values of color components can not be used for human detection since the resolution requirement is not met. Therefore, it is not suitable for low resolution and monochrome images.

The remaining of the paper is organized as follows: An overview of our previously proposed activity recognition and object detection system is given in section 2. Section 3 covers the proposed algorithm for human detection based on the eigenspace representation of human silhouettes (Figure 2). The results are displayed in section 4 while section 5 concludes the paper.

2. OVERVIEW

In our previous papers [6] and [7], we proposed a hierarchical method for human detection and activity recognition at different resolution levels. The first and second parts are object and activity detection requiring minimal decoding of compressed data. The last part is graph-based object detection in uncompressed domain. The proposed hierarchical scheme (Figure 1) enables working at different levels, from low complexity to low false rates. The first step is based on a principal component analysis of MPEG motion vectors to match the detected activity with known human activities; namely, walking, running, and kicking. The motion vectors are grouped automatically according to velocity, distance and human body proportions. The algorithm uses DC-DCT coefficients of the luminance and chrominance values when more detailed information is needed. These values are matched to activity templates and a human skin template (Figure 3). Detection of the head region from the skin color information is a crucial step for the performance of the matching algorithm. This requirement is not met for low resolution and monochrome JPEG images/MPEG I frames, and this leads us to the new algorithm, given in this paper, for human body detection (shaded block in Figure 1). The finest details in the sequences are obtained from the uncompressed domain via our proposed model based segmentation and graph matching algorithms [7]. The major contribution of the overall algorithm is to connect available data in compressed domain to high level semantics. For instance, consider a recorded video sequence taken from a fixed camera surveying a passage. The first step would retrieve possible frames where people walk. If a walking person is detected to stop, second step would analyze the extracted region for posture recognition. If a suspicious movement is detected, the third step would be a more detailed investigation of the region in the uncompressed domain.

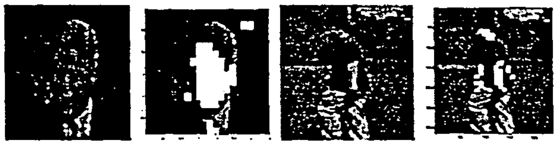


Figure 3: First and third: Original frames (YCbCr: 4:2:0 and 4:4:4), Second and fourth: Marked frames with macroblocks detected as skin regions.

3. DETECTION ALGORITHM

Our proposed human detection algorithm based on eigenspace representation of human silhouettes operates on the I-frames of MPEG video or JPEG images, using AC-DCT coefficients of image blocks. DCT compressed images encode a two-dimensional image using the DCT coefficients (c_{uv}) of an LxL image region ($I_{xy}, 0 \leq x < L, 0 \leq y < L$):

$$c_{uv} = K_u K_v \sum_{x=0}^{L-1} \sum_{y=0}^{L-1} I_{xy} cos \frac{\pi u(2x+1)}{2L} cos \frac{\pi v(2y+1)}{2L} \tag{1}$$

In Eq. 1, u and v denote the horizontal and vertical frequencies and $K_u = 1/\sqrt{L}$ if $u = 0$ and $K_u = \sqrt{2/L}$, otherwise. The AC components ($c_{uv}, u \neq 0$ or $v \neq 0$) capture the spatial frequency and directionality properties of the image block.

From the regenerated array of quantized coefficients, that are found during the JPEG decompression, the AC-DCT coefficients are extracted. Although they are quantized, the rank information is preserved and they can be used without any decoding procedure. The processing speed of the proposed method is fast since it does not require a fully decompressed MPEG video or JPEG image. The processing unit for the algorithm is a DCT block that is readily available from the compressed image.

To capture the intensity variations, first order AC coefficients (c_{01}, c_{10}, c_{11}) are used (Figure 5). DCT coefficient values capture the local directionality and coarseness of the spatial image. The vertical (horizontal) edges in uncompressed image correspond to high frequency component in the horizontal (vertical) frequencies and diagonal variations correspond to channel energies around the diagonal harmonics. Our approach is based on the observation that the structural information of human silhouettes can be captured from AC-DCT coefficients. In particular, the energy of blocks, that is obtained by summing up the absolute amplitudes of the first order harmonics, is used. The sides of the human body have a high response to the vertical harmonics while AC coefficients of the horizontal harmonics capture head, shoulder and belt lines (Figure 5). Furthermore, the corner edges at shoulders, hands and feet contribute to local diagonal harmonics. To train our system, 800 pedestrian images, obtained from the Artificial Intelligence Laboratory at MIT, are used. The pedestrians are centered in these 128x64 pixel windows. The windowing step in Figure 2 determines a 128x64 window and shifts it throughout the test image. The regions that have a lower AC energy than a given threshold (uniform regions), are eliminated. The following step resizes the image part in the 128x64 window to achieve multiscale detection. The scaling operation is done in compressed domain [9]. Note that the computational complexity of the transform domain manipulation techniques strongly depends on the number of zero DCT coefficients. Since the proposed algorithm uses three AC coefficients, the required computation can be further reduced by using sparse matrix multiplication techniques or other fast schemes in transformed domain [10].

Our goal is to find a compact representation of human silhouette by computing the principal components of the energy distribution of human bodies, or the eigenvectors of the covariance matrix of the human body images. These eigenvectors represent a set of features which together characterize the variation between human images. The number of eigenvectors (M) is equal to the number of images in the training set. In our algorithm we use the best eigenvectors ($M' = 12$) with the highest eigenvalues. Similarity measure in eigenspace representation for pattern matching in images is preserved under linear, orthogonal transformations. This implies that the principal component method gives exactly the same measure of match on transformed data as on pixel domain data. For lossy compression schemes such as JPEG and MPEG, the quantization of the transformed data is the cause for the degradation of the similarity measure.Although the DCT coefficients are quantized (furthermore, the coefficients except the three first order AC coefficients are quantized to zero), the essential information for matching purposes is preserved. The following steps summarize the recognition:

- Compute eigenvectors and eigenvalues from the training set of compressed human body images.
- Given an input image, calculate a set of weights based on the input image and the M' eigenvectors by projecting the input image onto each of the eigenvectors.
- Detect human regions by computing the distance between the mean adjusted input image and its projection onto human body space.

The training set of human images is $\Gamma_1, \Gamma_2, ..., \Gamma_M$, and the average is $\Phi = (\Gamma_1 + \Gamma_2 + ... + \Gamma_M)/M$. The difference of a human image from this average image is $\phi_i = \Gamma_i - \Phi$. Our goal is to find a set of M orthonormal vectors, $\mathrm{u_k}$ and their eigenvalues β_k which best describes the distribution of the data by using the principal component analysis. $\mathrm{u_k}$ and β_k are the eigenvectors and eigenvalues, respectively, of the covariance matrix C:

$$C = \frac{1}{M} \sum_{n=1}^{M} \phi_n \phi_n^T = AA^T \qquad (2)$$

where the matrix $A = \frac{[\phi_1 \phi_2 ... \phi_M]}{\sqrt{M}}$. The matrix C is a N by N matrix and the calculation of eigenvectors and eigenvalues of this matrix is a difficult task. To reduce the computational complexity, the eigenvectors x_k and eigenvalues λ_k of the matrix $A^T A$ are computed. It can be proven that the eigenvectors $\mathrm{u_k}$ of matrix C can be computed as [8]:

$$\mathrm{u_k} = \frac{\sum_{l=1}^{M} \phi_l x_{kl}}{\sqrt{\lambda_k}} \qquad (3)$$

and the eigenvalues are the same those matching x_k. The first 12 eigenimages obtained from 800 training images are shown in Figure 4. Creating the vector of weights for an image is equivalent to projecting the image onto the human body space. The distance ϵ between the image and its projection onto the body space is the distance between the mean adjusted input image $\phi = \Gamma - \Phi$ and $\phi_f = \sum_{k=1}^{M'} \omega_k \mathrm{u_k}$, its projection onto human body space, where $\omega_k = \mathrm{u_k^T}(\Gamma - \Phi)$ for $k = 1, ..., M'$.

4. RESULTS

The overall system performance is tested on 40 images and some of the human classification results are given in Figure 6 where windows with distance ϵ values smaller than a predefined threshold are displayed . The test images contain a total of 126 non-occluded frontal poses and the algorithm can detect 101 of them correctly.

The system is also trained for background classification by using several images where human is not present.

Our results are compared with those of [2] for frontal and near-frontal poses since our system is trained only for these view angles. The authors in [2] use an over-complete Haar dictionary of 16 x 16 pixels and train the system by using 564 positive examples that contain nonoccluded pedestrians and 597 negative examples that do not contain pedestrians. The detection rate for 141 nonoccluded pedestrian images in frontal or near-frontal images is 82%. In order to train our system, we use 800 positive examples and 600 negative examples with a bootstrapping algorithm. We achieve a correct detection rate of approximately 80%. Our approach has the advantage of using the available data in standard compression algorithms and gives highly accurate detection results.

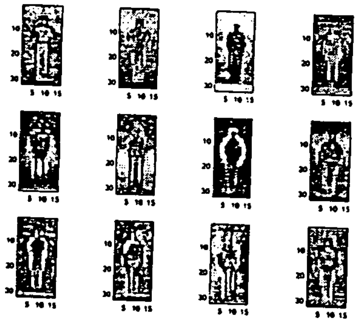


Figure 4: 12 eigenimages (upsampled).

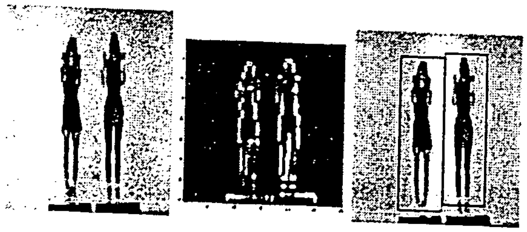


Figure 5: Left: Original image, Middle: AC-DCT values, Right: Classification result.

5. CONCLUSIONS

In this paper, the new algorithm block of our proposed activity recognition and human detection system is presented. Human detection in JPEG compressed still images and MPEG I frames is achieved by using AC-DCT coefficients. It is demonstrated that the structural information of human silhouettes in low resolution and monochrome images can be captured from AC-DCT coefficients. Our current work is to extend the algorithm for different views of human body images and for detection of other objects (e.g. cars).

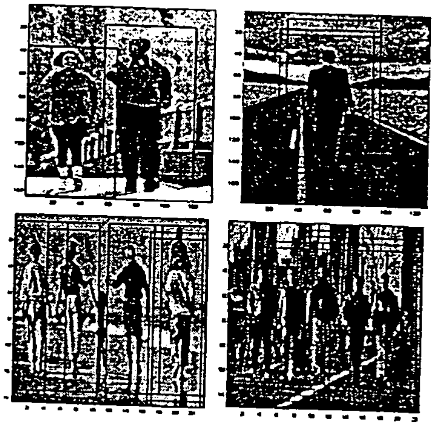


Figure 6: Human classification results.

6. REFERENCES

[1] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based Manipulation of Image Databases", International Journal of Computer Vision, 1996.

[2] C. P. Papageorgiou, M. Oren and T. Poggio, "Pedestrian Detection Using Wavelet Templates," *Proc. of CVPR*, Puerto Rico, June 1997.

[3] D. Schonfeld and D. Lelescu, "VORTEX: Video retrieval and tracking from compressed multimedia databases - template matching from MPEG2 video compressed standard", SPIE Conference on Multimedia and Archiving Systems III, Nov. 1998.

[4] Y. Zhong, H. Zhang, A. K. Jain, "Automatic Caption Localization in Compressed Video", IEEE PAMI, vol.22, no. 4, pp. 385-392, April 2000.

[5] H. Wang and Shih-Fu Chang, "A Highly Efficient System for Automatic Face Region Detection in MPEG Video Sequences", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 7, No. 4, pp. 615-628, Aug. 1997.

[6] I.B. Ozer, W. Wolf, A.N. Akansu, "Human Activity Detection in MPEG Sequences", IEEE Human Motion Workshop, Austin, Dec. 2000.

[7] I.B. Ozer, W. Wolf, A.N. Akansu, "Relational Graph Matching for Human Detection and Posture Recognition", SPIE Symposium on Voice, Video, and Data Communications, Boston, Nov. 2000.

[8] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces", CVPR 1991

[9] S. F. Chang and D. G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 1, pp. 1-11, Jan. 1995.

[10] R. Dugad, N. Ahuja, "A fast scheme for downsampling and upsampling in the DCT domain", ICIP 1999, pp 909-913.

A SMART CAMERA FOR REAL-TIME HUMAN ACTIVITY RECOGNITION

Wayne Wolf and I. Burak Ozer
Electrical Engr. Dept.
Princeton University
Princeton, NJ 08544, USA

Abstract - This paper describes a smart camera system under development at Princeton University. This smart camera is designed for use in a smart room in which the camera detects the presence of a person in its visual field and determines when various gestures are made by the person. As a first step toward a VLSI implementation, we use Trimedia processors hosted by a PC. This paper describes the relationship between the algorithms used for human activity detection and the architectures required to perform these tasks in real time.

1 INTRODUCTION

This paper describes the interplay between algorithms and architectures for a human activity recognition system. A number of groups, such as Olivetti Research, have developed **smart rooms** and buildings that track people. Early approaches used beacons carried by the subjects. However, a system that uses video avoids the need for beacons and allows the system to recognize gestures that can be used to command the operation of the smart room. A video-enabled smart room uses multiple **smart cameras** that both capture different views of the area and analyze the activity in their field of view. Tracking people and identifying what they are doing—walking, making gestures, etc.—is a challenging problem that requires the application of a number of different algorithms. The majority of research in human identification concentrates on algorithm development and is done in non-real time. Although Matlab is a powerful platform for algorithm development, Matlab scripts do not translate directly into efficient real-time implementations. We believe that developing real-time human activity recognition systems requires simultaneous study of algorithms and architectures. Architectural information generally places bounds on the amount of processing power available and may suggest that some types of algorithms are more efficient than others. Knowledge of the structure of the algorithms and data is essential to making the most of the available architectural resources. Our long-term goal is an integrated smart camera that includes a sensor, on-board processing, and on-board memory. We believe that heterogeneous multiprocessors are the most suitable architectures for smart cameras. In order to gain experience with

possible architectures, we are constructing a heterogeneous multiprocessor using a PC as a host and VLIW processors for video operations. This platform allows us to evaluate algorithms running on real-time data and to make measurements that would be too expensive to conduct using simulation.
This paper presents our prototype system for human activity recognition and presents our current view of the architectures that will be required for smart cameras. The next section summarizes previous related work. Section 3 briefly describes our PC-based testbed. Section 4 describes our algorithms for human activity recognition. Based on that algorithmic overview, Section 5 describes how the algorithms affect the underlying architecture.

2 PREVIOUS WORK

Great effort has been devoted to human recognition related topics such as face recognition in still images, and motion analysis of human body parts. Most of the previous work depend highly on the segmentation results and mostly motion is used as the cue for segmentation [1]. The major problems in the activity recognition are the scale, shift and projection changes between the model and the test data and segmentation dependency. A review of person identification, surveillance/monitoring. 2D/3D methods and smart rooms can be found in [2] where occlusion and resolution problems are pointed out suggesting the use of multiple cameras. In addition to MIT Media Lab [3], [4], a similar research is conducted in the University of Maryland Keck Laboratory for the analysis of visual motion where multiple cameras are attached to a network of sixteen PCs used for both data collection and real time video analysis [5]. Watlington and Bove [6] developed a dataflow architecture for real-time video processing. Wandell et al. [7] developed a sensor array that can be programmed to provide different capture characteristics around the array. Foote and Kimber [8] built a panoramic camera system built from multiple cameras.

3 TESTBED ARCHITECTURE

Our testbed architecture is a heterogeneous multiprocessor. We use two Trimedia processors on two PCI cards attached to a host PC. Each Trimedia evaluation board includes a TM32 processor, local memory, and analog video input and output. Most video operations are performed on the on-board memory. The TM32 can also talk to the host PC using PCI transfers. The TM32 is programmed using the Trimedia C compiler running on the host PC. The Trimedia evaluation board is designed to support multiprocessing. TM32s can communicate via shared memory using the on-board memories without communicating directly with the host.

4 ALGORITHMS

We have developed a hierarchical method for human detection and activity recognition at different resolution levels (Figure 1). This methodology was originally developed using Matlab; we are now adapting it for real-time

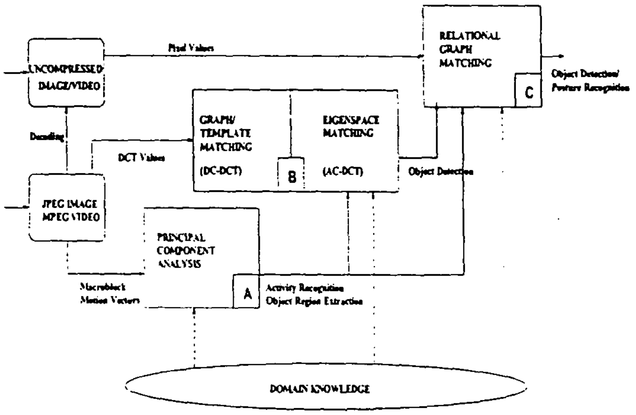


Figure 1: Algorithm Blocks for Human Detection and Activity Recognition

multiprocessing. The first and second parts are object and activity detection requiring minimal decoding of compressed data. The last part is graph-based object detection in uncompressed domain. The proposed hierarchical scheme enables working at different levels, from low complexity to low false rates. Since the data is stored in MPEG video format, this hierarchical scheme provides a more effective retrieval scheme for both online and off-line applications. For instance, consider a recorded video sequence taken from a fixed camera surveying a passage. The first step will retrieve possible frames where people walk. If a walking person is detected to stop, second step will analyze the extracted region for posture recognition. If a suspicious movement is detected, the third step will be a more detailed investigation of the region in the uncompressed domain.

Block A: Activity Detection by Using MPEG Motion Vectors: The major contribution of this algorithm is to connect available data in compressed domain to high-level semantics. Since the resolution of the motion vectors is one macroblock and there is no direct correspondence with the object parts and their motion, a robust and global model must be used. Principal component analysis (PCA) method is one of the global approaches. PCA has been successfully used by Yacoob and Black [9] for human activity recognition in uncompressed video sequences where the motion measurements for segmented human body parts are used. In our method, the measurements correspond to macroblock groups corresponding to human region. This step is based on a principal component analysis of MPEG motion vectors to match the detected activity with known human activities; namely, walking, running, and kicking [10]. The motion vectors are extracted by using the MPEG2 encoder and video support libraries of Trimedia TM-1300 processor. The recognition part can then be divided into two subparts: a) grouping motion vectors, b) principal component analysis of motion groups.

a) Grouping motion vectors: The motion vectors are grouped automatically according to velocity, distance and human body proportions. For each P-

frame, let N=v1*v2 be the number of motion vectors (v1=frame width/16 and v2=frame height/16). We group these vectors by using a connected component algorithm. The time complexity of a serial connected component algorithm is O(v1v2) for an v1 x v2 macroblock frame. The object is segmented to three parts (upper body, torso and lower body) according to the human body proportions. For training the system, several sequences which are temporally aligned are used. The mean of the motion vectors in horizontal and vertical direction is computed for the macroblocks corresponding to each part (6 parameters) for a number of sequences T. Let A be a matrix of dimensions $6T \times k$, formed by the training set of k different examples for each activity. The singular value decomposition of the matrix A ($A = U\Sigma V^T$) is computed to get the approximated projection of the exemplar vectors (columns of A) onto the subspace spanned by the $q < k$ basis vectors. Hence activity basis with parameters $m = A'U$ are computed. This process is done off-line.
b) To recognize the activities, a motion group of an activity which can be shifted and scaled in time is compared with the training set. Let $[D]_j$ denote the j-th element of vector $[D]$ of an observed activity that can be scaled and shifted. By projecting this vector on the activity basis, a coefficient vector, $\bar{c}$, is recovered, which approximates the activity as a linear combination of activity basis. For recovering the coefficients, an error function ρ has to be minimized to find the coefficient vector as $E(\bar{c}) = \sum_{j=1}^{nT} \rho(([D]_j - \sum_{l=1}^{q} c_l U_{l,j}), \sigma)$. The normalized distance ($d^2 = \sum_1^q (c_i/||c|| - m_i/||m||)^2$) between the coefficients m_i from the training data set and coefficients of exemplar activities c_i is used to recognize the observed activity. 10 training test sequences for each activity class are obtained from various sources for the side-view. The camera motion is assumed to be zero. Table 1 displays the resulting average values of the normalized distances between the activity sets and test sequences. The last sequence is a MPEG car movie where the distances are very high for each activity class.

	Walking	Running	Kicking
Walking	0.0153	0.0956	0.1383
Running	0.4286	0.0456	0.1970
Kicking	0.244	0.1172	0.0722
car	0.5362	0.4282	0.6922

Table 1: The average values of the normalized Euclidean distances between the activity sets and test sequences.

Block B: Human Detection in the Compressed Domain: In this algorithm, the overall shape of a standing or walking person (from front or back-view) in MPEG I frames is detected by using the AC-DCT coefficients. Most of the retrieval systems that are based on the compression schemes are devised for particular objects. This differentiates our approach from previous work where the compression algorithms are determined by characteristics of object of interest to be retrieved. DCT coefficient values capture the local directionality and coarseness of the spatial image. Our approach is based on the observation that the structural information of human silhouettes can be captured from AC-DCT coefficients. We trained our system with 800 pedestrian images obtained from the Artificial Intelligence Laboratory at MIT. The pedestrians are centered in these 128x64 pixel windows. The regions that have a lower AC energy than a given threshold (uniform regions) are eliminated. The following step resizes the image part in the 128x64 window to achieve multiscale detection. The scaling operation is done in compressed domain [11]. Note that the computational complexity of the transform domain manipulation techniques strongly depends on the number of zero DCT coefficients. Since the proposed algorithm uses three AC coefficients, the required computation can be further reduced by using sparse matrix multiplication techniques or other fast schemes in transformed domain. To capture the intensity variations, first order AC coefficients, obtained from 8x8 DCT blocks, are used. The AC image, extracted from the MPEG I-frame, that is obtained by summing these three coefficients is the input of the detection algorithm [12]. The training set of human images is $\Gamma_1, \Gamma_2, ..., \Gamma_M$, and the average is $\Phi = (\Gamma_1 + \Gamma_2 + ... + \Gamma_M)/M$. The difference of a human image from this average image is $\phi_i = \Gamma_i - \Phi$. Our goal is to find a set of M orthonormal vectors, $\mathbf{u_k}$ and their eigenvalues $\beta_\mathbf{k}$ which best describes the distribution of the data by using the principal component analysis. $\mathbf{u_k}$ and $\beta_\mathbf{k}$ are the eigenvectors and eigenvalues, respectively, of the covariance matrix $C = \frac{1}{M}\sum_{n=1}^{M} \phi_n \phi_n^T$. Note that these computations are done off-line. A given image window of size of the training images is scaled in the compressed domain to form AC input image. The distance ϵ between the AC image and its projection onto the body space is the distance between the mean adjusted input image $\phi = \Gamma - \Phi$ and $\phi_f = \sum_{k=1}^{M'} \omega_k \mathbf{u_k}$, its projection onto human body space, where $\omega_k = \mathbf{u_k^T}(\Gamma - \Phi)$ for $k = 1, ..., M'$. The overall system performance was tested on 40 images. We achieve a correct detection rate of approximately 80%. The input of the graph matching algorithm is pixel values in the uncompressed domain or 8x8 block values in compressed domain. It is explained in the next section for uncompressed domain.

Block C: Graph Matching: This block has four components: Detection of skin areas and foreground objects, contour extraction of connected components, parametric modeling (superellipse fitting) and graph matching [14]. Skin areas are detected by comparing color values to a human skin model. In our previous work [14], we have used Farnsworth nonlinear transformation in order to obtain uniform circular color differences. However, prior knowledge about the camera system and background increases the robustness of simpler skin color models suitable for real-time applications. We use YUV color model where chrominance values are downsampled by two. Then we search the image for a connected component (skin area) by scanning the image and we apply the contour following algorithm that uses the 3x3 filter to follow the edge of the component where the filter can move in any of 8 directions to follow the edge . Each contour of size c_i is then fitted to a superellipse with 6 parameters by a Levenberg-Marquardt minimization method [15]. The results are possible head regions. Then for each head region with extracted foreground segments fitted to superellipses, graph matching is performed. Face detection allows to start initial branches efficiently and reduces the complexity. Note that false face detection will result in a branch with single or very few matched nodes and will be eliminated. Each body part and meaningful combinations represent a class (ω) where the combination of binary and unary features are represented by a feature vector (X) and computed off-line. Note that feature vector elements of a frame node computed online by using superellipse parameters, change according to body part and the nodes of the branch under consideration. For example, for the first node of the branch, feature vector consists of unary attributes. The feature vector of the following nodes includes also binary features dependent on the previous matched nodes in the branch. For the purpose of determining the class of these feature vectors a piecewise quadratic Bayesian classifier is used. The computations needed for each node matching are then a function of the feature size and the previously matched nodes of the branch under consideration. The marked regions are tracked by using superellipse parameters for the consecutive frames and graph matching algorithm is applied for new objects appearing in the other regions.

5 INTERPLAY BETWEEN ALGORITHMS AND ARCHITECHTURES

The algorithms of color segmentation, contour following of connected components, superellipse fitting and graph matching are implemented on a Pentium III processor (500 MHz) and on the TM32 processors separately. The frame size is 384x240. Figure 2 shows example frames from the original sequences. The processing time for each algorithm block on one TM32 processor is displayed in Figure 3. The processing times for each algorithm block on Pentium III processor are between 50-60 msecs for skin detection, 65-80 msecs for contour following, 180-400 msecs for superellipse fitting and 10-60 msecs for graph matching algorithm. We use MPEG2 encoder on a TM32 processor where the processing time for P frame coding using motion vectors is between 170-215 msecs and the processing time for coding of the DCT coefficients of intra and inter frames is between 45-65 msecs.

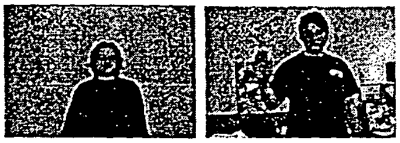


Figure 2: Frames from the original test sequences.

Our algorithmic pipeline clearly performs a wide range of disparate operations: 1) pixel-by-pixel operations, such as color segmentation; 2) pixel-region operations, such as region identification; 3) mixed operations, such as superellipse fitting; 4) non-pixel operations, such as graph matching.
We start with operations that are clearly signal-oriented and move steadily away from the signal representation until the data is very far removed from a traditional signal representation.

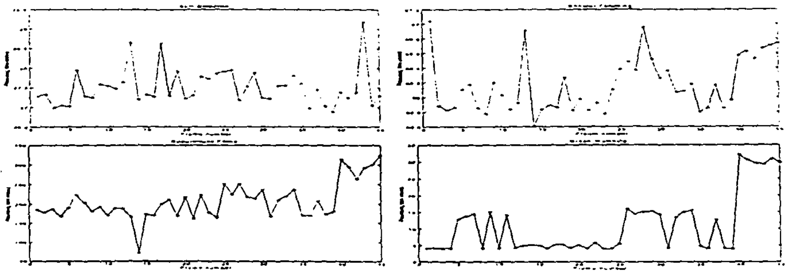


Figure 3: Processing times for skin detection, contour following, superellipse fitting and graph matching.

We face two basic questions when designing the architecture of a multiprocessor for this problem: what types of processors to use and how they are to communicate. Clearly, heterogeneous processing is called for because different processor architectures are well-suited to different stages of the processing pipeline. VLIW architectures are very well-suited to pixel-oriented operations. A RISC architecture may be a less expensive engine for later steps such as graph matching. We expect VLIW architectures to be used in the front-end steps and RISC architectures to be used for some of the later steps. Communication is also somewhat heterogeneous and fairly predictable. Information in our system flows primarily forward. While the human visual system clearly uses some feedback, the amount of information flowing backward is small relative to the forward information flow. As a result, traditional shared memory processors used for scientific computing are not necessarily the best option. Each processor does not need a full view of the memory space. Instead, each processor needs to receive data from its predecessor and pass on its results to the next processor. In general, the volume of data goes down as image processing progresses. We therefore propose a **macropipeline architecture**. Processors handling adjacent steps will have a shared memory space so that data can be passed between them. Only a small amount of globally shared memory is required for coordinating the processors. The Trimedia evaluation boards can support this model using their shared memory mechanism, although they also allow more general shared memory. A custom VLSI implementation would be able to use simpler memory interconnection networks to provide the more localized shared memory.

6 CONCLUSIONS

Smart camera design is an exciting challenge for VLSI signal processing systems. Human activity recognition is a complex task that is ideally suited to VLSI implementation due to the performance benefits of keeping data flows within a single chip. Our experiments in adapting human activity recognition algorithms to real-time show that heterogeneous processor architectures make sense to capture the nature of data flow and computations. Our experience also shows that this problem has a natural data flow structure analogous to the pixel-level data flow within a single video algorithm. We plan to use experience with our PC-and-VLIW multiprocessor system to help us design a single-chip smart camera.

ACKNOWLEDGMENTS

This work was supported by the New Jersey Center for Pervasive Information Technology with equipment support from Trimedia Technologies.

References

[1] J. K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review," Computer Vision and Image Understanding, vol. 73, no. 3, pp. 428-440. March 1999.

[2] A. Pentland, "Looking at People: Sensing for Ubiquitous and Wearable Computing", IEEE PAMI, Vol 22, No 1, pp. 107-119, Jan. 2000.

[3] A. Pentland, T. Choudhury, "Face Recognition for Smart Environments", Computer , Vol. 33 Issue 2 , pp. 50-55, Feb. 2000.

[4] A. D. Wilson, A. F. Bobick, "Realtime Online Adaptive Gesture Recognition", International Conference on Pattern Recognition, pp. 270-275, 2000.

[5] L. S. Davis, Eugene Borovikov, Ross Cutler, and Thanarat Horprasert,"Multiperspective Analysis of Human Action", Third International Workshop on Cooperative Distributed Vision, 1999.

[6] J. Watlington and V. M. Bove, Jr., "A System for Parallel Media Processing," Parallel Computing, 23:12, December 1997.

[7] B. Wandell, P. Catrysse, J. DiCarlo, D. Yang and A. El Gamal, "Multiple Capture Single Image Architecture with a CMOS SEnsor", International Symposium on Multispectral Imaging and Color Reproduction for Digital Archives (Society of Multispectral Imaging of Japan), Chiba, Japan, pp. 11-17, 1999.

[8] J. Foote and D. Kimber, "FlyCam: Practical Panoramic Video and Automatic Camera Control," IEEE International Conference on Multimedia and Expo, pp. 1419-1422, 2000.

[9] Y. Yacoob and M. J. Black, "Parameterized Modeling and Recognition of Activities", ICCV, pp.120-127, 1998.

[10] B. Ozer, W. Wolf, Ali N. Akansu, "Human Activity Detection in MPEG Sequences", Proceedings of IEEE Workshop on Human Motion, Austin, pp. 61-66, December 2000.

[11] S. F. Chang and D. G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video," IEEE Journal on Selected Areas in Communications, vol. 13, no. 1, pp. 1-11, Jan. 1995.

[12] B. Ozer, W. Wolf, "Human Detection in Compressed Domain", to appear in ICIP 2001.

[13] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces", CVPR, pp. 586-591, 1991.

[14] B. Ozer, W. Wolf, A. N. Akansu, "Relational Graph Matching for Human Detection and Posture Recognition", SPIE, Photonic East 2000, Internet Multimedia Management Systems, Boston, November 2000.

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical Recipes in C ", Cambridge University Press, Second Edition, 1995.

Video Analysis for Smart Rooms

I. Burak Ozer and Wayne Wolf
Department of Electrical Engineering, Princeton University
Princeton, NJ 08544, USA

ABSTRACT

Smart rooms provide advanced interfaces for networked information systems. Smart rooms include a variety of sensors that can analyze the behavior of persons in the room; these sensors allow people to issue commands without direct contact with equipment. Video is one important modality for smart room input—video analysis can be used for determining the presence of people in the room, gesture analysis, facial analysis, etc. This paper outlines the architecture of a real-time video analysis system for smart rooms. The system uses multiple cameras, each with its own video signal processor (VSP). We use algorithms that can be performed in real-time to capture basic information about the persons in the room.

Keywords: Real-time video analysis, smart rooms, human and activity detection

1. INTRODUCTION

Smart rooms use a variety of sensors to monitor activity in the room. This paper describes our work on smart cameras for smart rooms. Smart cameras provide the eyes for smart rooms by providing real-time visual recognition. Smart cameras can be used to detect, recognize, and analyze people or other objects in the room. The results of smart camera analysis can be used (often in concert with information from audio and other modalities) to control the operation of devices in the room, detect the presence of unwanted people, or a variety of other tasks. We have developed algorithms for the real-time recognition of persons and classification of their activities. After identifying a person in the room, we classify that person's pose by fitting shapes to the person's body parts and modeling the relationships between the shapes in a graph. We then compare the constellation of shapes against models in a library. Our algorithm can identify a variety of poses by a person at a rate of several poses per second. By tracking the person's activity as expressed by poses, smart cameras can provide information about the person for use by higher-level systems of the smart room. CMOS image sensors are becoming both better and less expensive. Advances in VLSI technology also allow us to put a powerful video processor on a single chip. The video processor may be on the same chip as the image sensor or may be put on a separate chip, depending on the requirements of circuit design. The image sensor and processor together form a smart camera node. Several groups have developed smart camera systems. Watlington and Bove[1] developed a dataflow architecture for real-time video processing. Wandell et al.[2] developed a sensor array that can be programmed to provide different capture characteristics around the array. Foote and Kimber[3] built a panoramic camera system built from multiple cameras. Nicolescu and Medioni[4] use image processing algorithms to electronically pan, tilt, and zoom through images supplied by an array of cameras. Chai et al.[5] developed an architecture for pixel-level processing in the imaging array. A review of person identification, surveillance/monitoring, 2D/3D methods and smart rooms can be found in[7] where occlusion and resolution problems are pointed out suggesting the use of multiple cameras. An extensive research has been done at MIT Media Lab on smart rooms for several applications.[8,9] A similar research is conducted in the University of Maryland Keck Laboratory for the analysis of visual motion where multiple cameras are attached to a network of sixteen PCs used for both data collection and real time video analysis.[10]

The next section describes the architecture of our prototype smart camera system and our projected VLSI architecture. Section 3 describes our algorithms for person recognition and activity classification. Section 4 summarizes results from our system.

2. ARCHITECTURE

Our prototype smart camera is based on a PC platform. We use a standard camera that provides NTSC composite video. The host PC includes a pair of Trimedia TM-1300 video processing boards. Each video processing board includes a five-issue VLIW processor. The Trimedia processors and the host PC communicate through shared memory. At this writing, our programs run on a single Trimedia processor. We are redesigning the programs to run on the multiprocessor.

We have previously described characteristics of a VLSI smart camera.[6] We are now conducting a series of experiments to better characterize the computational requirements of human activity analysis and the best architecture to implement such algorithms. We expect the system to be a loosely coupled multiprocessor that includes both VLIW and RISC processors. We expect that each processing node will include both shared and dedicated memory.

3. ALGORITHMS

The algorithm blocks are shown in Figure 1: Background elimination, detection of skin areas and foreground objects, contour extraction of connected components, parametric modeling (superellipse fitting) and graph matching.[11]

- Background elimination and color transformation: First step is the transformation of pixels (m by n) into another color space regarding to the application. For example transforming the RGB values into YUV components takes 5 additions and 8 multiplications for each pixel. Background elimination is performed by using these transformed pixel values. Our assumption for the background elimination is that the background is known and there is no change in the lighting conditions during the whole test sequence.
- Skin area detection: Skin areas are detected by comparing color values to a human skin model. In our previous work,[11] we have used Farnsworth nonlinear transformation in order to obtain uniform circular color differences. However, prior knowledge about the camera system and background increases the robustness of simpler skin color models suitable for real-time applications. We use YUV color model where chrominance values are downsampled by two.

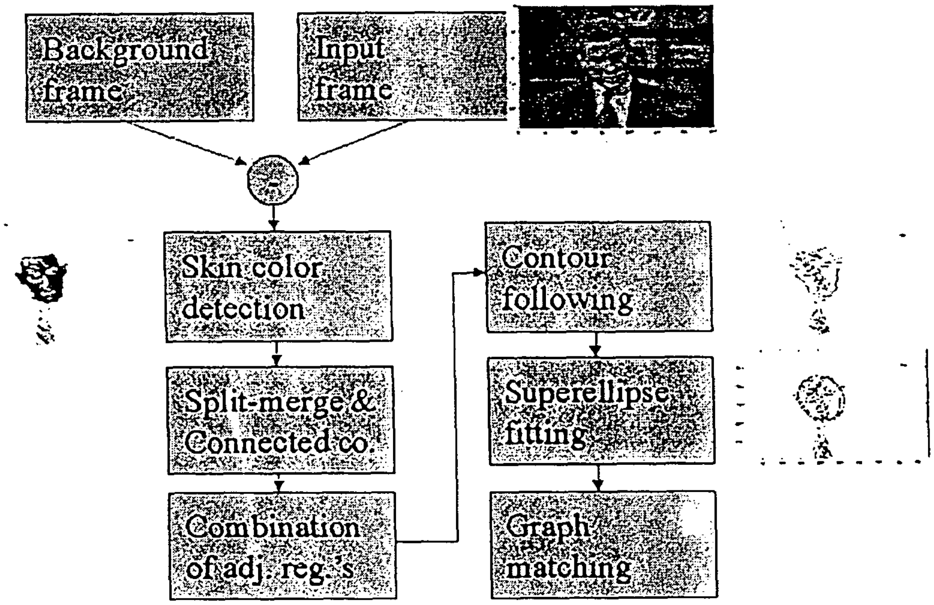


Figure 1. Algorithm blocks and corresponding results of selected steps for head localization.

- Segmentation of non-skin areas and connected component algorithm: An object usually contains several subobjects that can be obtained by segmenting the object of interest hierarchically into its smaller unique parts. The foreground regions that are adjacent to detected skin areas are extracted and corresponding connected components are found. We combine the meaningful, adjacent segments and use them as the .input of the following algorithm steps.

- Contour following: We apply the contour following algorithm that uses the 3x3 filter to follow the edge of the component where the filter can move in any of 8 directions to follow the edge . Each contour of size c_i is then fitted to a superellipse with 5 parameters by a Levenberg-Marquardt minimization method.[12]

- Superellipse fitting: Even when human body is not occluded by another object, due to the possible positions of non-rigid parts a body part can be occluded in different ways. For example, hand can occlude some part of torso or legs. In this case, 2D approximation of parts by fitting superellipses with shape preserving deformations provides more satisfactory results. It also helps to discard the deformations due to the clothing. Global approximation methods give more satisfactory results for human detection purposes. Hence, instead of region pixels, parametric surface approximations are used to compute shape descriptors.

  The inside-outside function of a superellipse can be given as:

$$(\frac{x}{a_x})^{2/\epsilon} + (\frac{y}{a_y})^{2/\epsilon} = f(x, y, \mathrm{a}) \qquad (1)$$

  where a is the parameter set.

  First, the initial parameter set is used to find non-deformed world centered superellipse $(\overline{x}, \overline{y})$ where $(D \circ R \circ T)^{-1}(X, Y) \rightarrow (\overline{x}, \overline{y})$ with D =Deformation, R =Rotation, T =Transformation. The model to be fitted, the inside-outside function $f(\overline{x}, \overline{y}, \mathrm{a})$ forms the merit function χ in order to determine best fit parameters by its minimization. With nonlinear dependences, the minimization must proceed iteratively. The procedure is repeated until χ^2 stops decreasing.

- Graph matching: Each extracted region modeled with ellipses correspond to a node in the graphical representation of human body. Face detection allows to start initial branches efficiently and reduces the complexity. Each body part and meaningful combinations represent a class (ω) where the combination of binary and unary features are represented by a feature vector (X) and computed off-line. Note that feature vector elements of a frame node computed online by using superellipse parameters, change according to body part and the nodes of the branch under consideration. For example, for the first node of the branch, feature vector consists of unary attributes. The feature vector of the following nodes includes also binary features dependent on the previous matched nodes in the branch. For the purpose of determining the class of these feature vectors a piecewise quadratic Bayesian classifier with discriminant function $g(X)$ is used. The generality of the reference model attributes allows the detection of different postures while the conditional rule generation (r) decreases the rate of false alarms. The computations needed for each node matching are then a function of the feature size and the previously matched nodes of the branch under consideration. The marked regions are tracked by using superellipse parameters for the consecutive frames and graph matching algorithm is applied for new objects appearing in the other regions. The overall algorithm for the relational graph matching is given below to match model graph nodes $(j \in O_r)$ to image graph nodes $i \in O_n$.

```
for every model node $j \in O_r$ do
    for every branch $b$ do
     $(i_1, i_2)$ =match$(j, b)$
     copy branch $b$ and add node pair $(j, i_1)$ in the new branch and update $G^b$ by adding $g_j^b(X_{i_1})$
     copy branch $b$ and add node pair $(j, i_2)$ in the new branch and update $G^b$ by adding $g_j^b(X_{i_2})$
    end for
end for
choose $arg\,\max_{b \in B_h} G^b$
```

```
match($j, b$)
for every image node $i \in O_n$ do
    for every matched node pair $(b_j, b_i)$ in the branch do
        if $\exists\, r(b_j, j)$ then
            if $r(b_i, i)$ holds then
                compute $g_j^b(X_{i,b_i})$
            else
                $g_j^b(X_{i,b_i}) = 0$
            end if
        else
                compute $g_j^b(X_i)$
        end if
    end for
end for
Return image nodes $i_1, i_2$ with two highest $g_j^b(x_i)$ values > threshold
```

4. RESULTS

The emphasis of this paper is on the real-time applicability of our human detection and activity recognition method that we have developed for off-line annotation of digital libraries. Our main goal is to gain experience with possible architectures. For this purpose, we investigate the processing time of the algorithm blocks by testing and modifying them on TM1300 video processing board. At present, we use a Trimedia processor on a PCI card attached to a host PC. The Trimedia evaluation board includes a TM32 processor, local memory, and analog video input and output. Most video operations are performed on the on-board memory. The TM32 can also talk to the host PC using PCI transfers. The TM32 is programmed using the Trimedia C compiler running on the host PC.

The processing time for each algorithm block is displayed in Figures 3. Figure 2 shows example frames from the original sequences. Some snapshots of the sequence for each algorithm block are displayed in Figure 4. The frame size is 384x240. The preliminary results show that 82 % of the body parts are correctly classified. The remaining 18 % is the miss detection. Note that, an effective parallel implementation for some parts of the algorithm would improve the processing time, e.g, ellipse fitting, branch evolution in graph matching. The Trimedia evaluation board is designed to support multiprocessing. TM32s can communicate via shared memory using the on-board memories without communicating directly with the host. Our current work includes to test the algorithms using two processors.

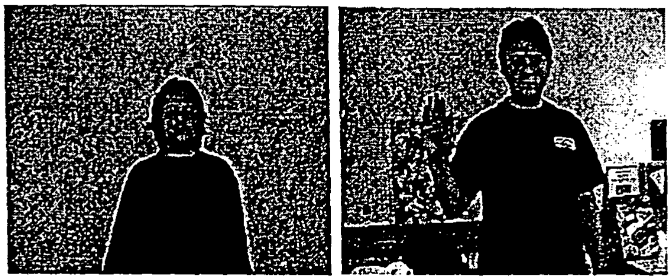


Figure 2. Frames from the original test sequences.

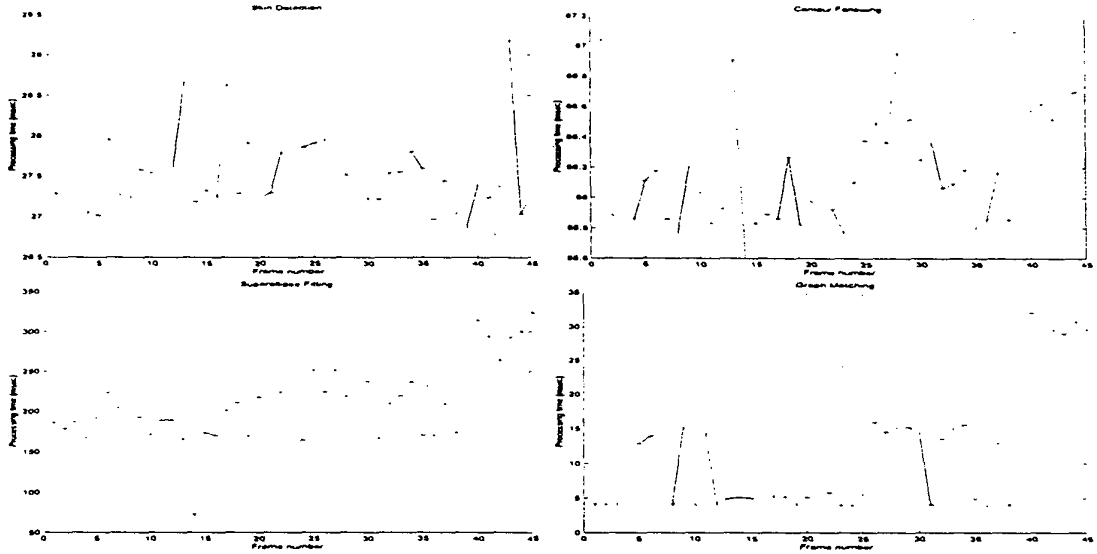


Figure 3. Processing times for skin detection, contour following, superellipse fitting and graph matching.

We face two basic questions when designing the architecture of a multiprocessor for this problem: what types of processors to use and how they are to communicate. Clearly, heterogeneous processing is called for because different processor architectures are well-suited to different stages of the processing pipeline. VLIW architectures are very well-suited to pixel-oriented operations. While they perform reasonably for problems like graph matching, a RISC architecture is arguably equally well-suited to the problem. We expect VLIW architectures to be used in the front-end steps and RISC architectures to be used for some of the later steps.

Traditional shared memory processors used for scientific computing are not necessarily the best option. Each processor does not need a full view of the memory space. Instead, each processor needs to receive data from its predecessor and pass on its results to the next processor. In general, the volume of data goes down as image processing progresses. We therefore propose a **macropipeline architecture**. Processors handling adjacent steps will have a shared memory space so that data can be passed between them. Only a small amount of globally shared memory is required for coordinating the processors. The Trimedia evaluation boards can support this model using their shared memory mechanism, although they also allow more general shared memory. A custom VLSI implementation would be able to use simpler memory interconnection networks to provide the more localized shared memory.

5. SUMMARY

Advances in VLSI technology will allow us to deploy smart cameras that can analyze activity in a room in real time. The smart rooms that we build with such smart cameras can be used to augment human potential, to monitor activity in the room, and for many other tasks. Today's processing power allows us to analyze poses of humans in real time with relatively modest hardware. Our algorithms use graph models to describe the posture of a person in the frame and to compare that posture against known poses.

ACKNOWLEDGMENTS

This work was supported by the New Jersey Center for Pervasive Information Technology and by the National Science Foundation.

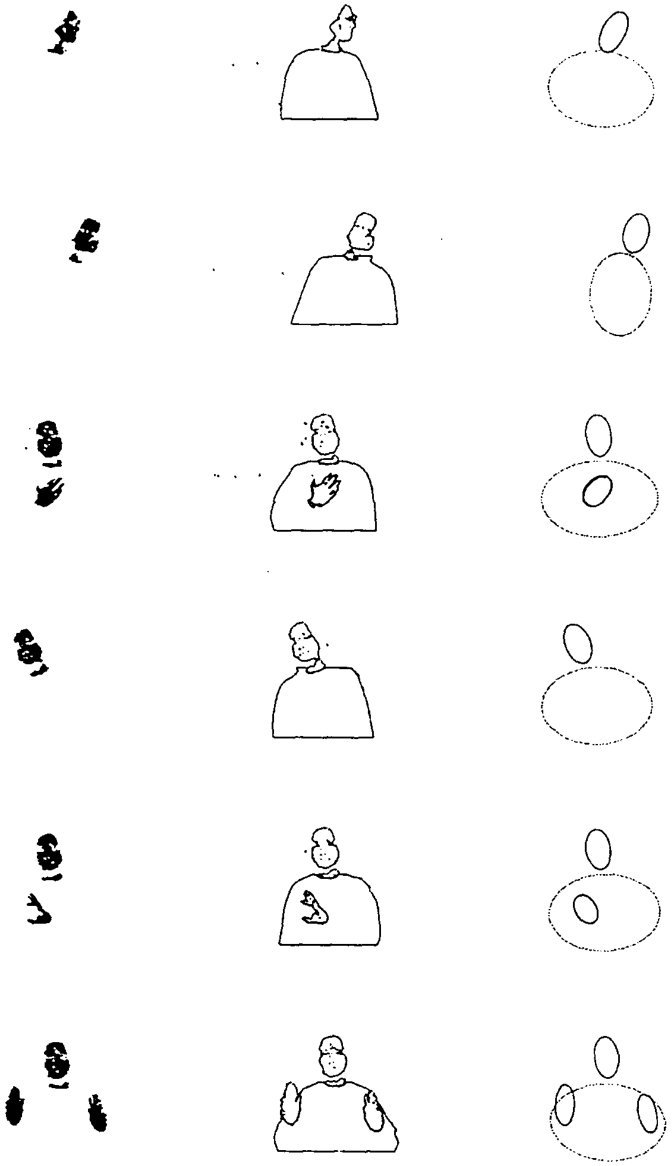


Figure 4. Snapshots of the sequence for each algorithm block. The correct classification percentage of head, torso and hands is 82 percent. Left: Skin color detection. Middle: Contour following. Right: Superellipse fitting.

REFERENCES

1. J. Watlington and V. M. Bove, Jr., "A System for Parallel Media Processing," Parallel Computing, 23:12, December 1997.
2. Wandell, Catrysse, DiCarlo, Yang and El Gamal (1999). In Proceedings of the International Symposium on Multispectral Imaging and Color Reproduction for Digital Archives. Chiba, Japan. October 21- 22. P. 11-17. Society of Multispectral Imaging of Japan.
3. Jonathan Foote and Don Kimber, "FlyCam: practical panoramic video and automatic camera control," in Proceedings, 2000 International Conference on Multimedia and Expo, IEEE, 2000.
4. Mircea Nicolesceu and Gerard Medioni, "Electronic pan-tilt-zoom: a solution for intelligent room systems," in Proceedings, 2000 International Conference on Multimedia and Expo, IEEE, 2000.
5. S. M. Chai, A. Gentile, W. E. Lugo-Beauchamp, J. Fonseca, J. L. Cruz-Rivera, and D. S. Wills, "Focal Plane Processing Architectures for Real-time Hyperspectral Image Processing," Applied Optics, Special Issue on Optics in Computing, 39:(5), pages 835-849, February 2000.
6. Wayne Wolf, "VLSI distributed architectures for smart cameras," in Media Processors 2001, Proceedings of SPIE Vol. 4313, SPIE, 2001.
7. A. Pentland, "Looking at People: Sensing for Ubiquitous and Wearable Computing", IEEE PAMI, Vol 22, No 1, pp. 107-119, Jan. 2000.
8. A. Pentland, T. Choudhury, "Face Recognition for Smart Environments", Computer , Vol. 33 Issue 2 , pp. 50-55, Feb. 2000
9. A. D. Wilson, A. F. Bobick, "Realtime Online Adaptive Gesture Recognition", International Conference on Pattern Recognition, pp. 270-275, 2000.
10. L. S. Davis, Eugene Borovikov, Ross Cutler, and Thanarat Horprasert,"Multi-perspective Analysis of Human Action", Third International Workshop on Cooperative Distributed Vision, 1999.
11. Burak Ozer, Wayne Wolf, Ali N. Akansu, "Relational Graph Matching for Human Detection and Posture Recognition", SPIE, Photonic East 2000, Internet Multimedia Management Systems, Boston, November 2000.
12. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical Recipes in C ", Cambridge University Press, Second Edition, 1995.

Workload Characterization for Smart Cameras

Tiehan Lv, I. Burak Ozer and Wayne Wolf
Department of Electrical Engineering, Princeton University
Princeton, NJ 08544, USA

Abstract

As part of the research for developing a smart camera system, we conducted a workload characterization study. The target program represents an emerging catalog of video applications in that it encompasses a broad range of operations from low-level pixel operations to high-level abstract objects matching. In this work, we analyze the instruction statistics, branch behavior and memory access behavior of the target program. Manual optimization on the target program is performed to exploit the instruction level features of TriMedia system. In addition to micro-architecture level analysis, we also discuss the issues of parallel structure for multiprocessor.

1 Introduction

With the progress of the information technology, video analysis involving humans becomes one of the most active domains in computer vision. As an application of video analysis, smart cameras are video cameras with their own video-processing elements. Smart cameras can be used to recognize people and their activities[1]. The application developed for smart cameras consists a broad range of widely used image processing techniques such as point processing, image segmentation and pattern recognition, which makes the application a representative for the emerging catalog of video analyzing applications.

It is essential for us to understand the application behavior to develop efficient hardware for a smart camera system. Decisions such as the number of processors in the system, the topology of the processors, cache parameters of each processor, the number of ALUs, ISA (instruction set architecture), etc. all rely on the characteristic of the application running in the system. In this work, we use SimpleScalar tool set[4] and TriMedia develop tool kit[5] to examine the behavior of the algorithm used in smart camera system. While this part of work is similar to the approach described in [3], we focus only on the specific algorithm for a smart camera system.

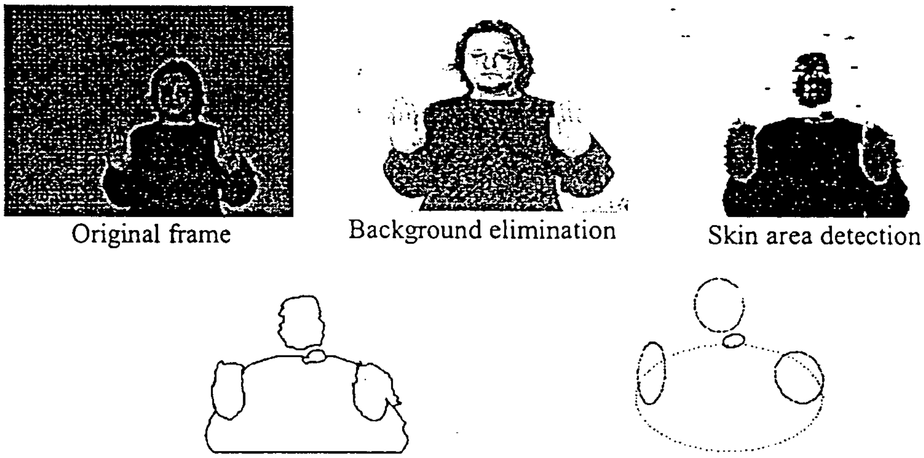
Although the complicated modern compilers perform quite efficient and complex optimization for applications, manual optimization on specific algorithms can still bring a considerable speedup. In addition, manual optimization can help identifying those most effective hardware features so that these features can be kept in the next design. Furthermore, effective manual optimization methods can be integrated into compilers. For these reasons, we perform manual optimization for the smart camera algorithm and the effect is evident—the running time is reduced by more than 80%.

Since video processing needs intensive computation, multiple processors may be required by the system. In this case, the organization of the processors in the system becomes an important issue. To discuss this, we evaluate the communication cost of data transference and synchronization operation in experiment. Then, two different parallel processing architectures for smart cameras are compared.

The remainder of the paper is organized as follows. Section 2 describes the target algorithm. Section 3 introduces the evaluation environment. Section 4 describes the characterization of the proposed algorithm. Section 5 describes the manual optimization of the algorithm. Section 6 discusses two different multi-processor architectures. Section 7 presents the conclusion.

2 Algorithm description

The target algorithm, which is developed by Ozer et al [2], can recognize people and their body parts. Thus, a computer is able to recognize a person's poses and activities. The algorithm consists of several parts—background elimination, skin area detection, contour following, super ellipse fitting and graph matching. Figure 1 shows a frame processed by each phase.



Original frame Background elimination Skin area detection

Contour following Superellipse fitting

Figure 1: A frame processed by target program

♦ *Background elimination*:

The background elimination phase takes 384x240 images as input and extracts foreground objects from the input image. The output is used by skin area detection. We assume that background is known a prior, since a smart camera is supposed to operate in a relatively fixed environment. With this assumption, this pixel level process simply subtracts background from foreground image. The output image containing foreground objects are then processed by next stage.

♦ *Skin area detection*:

This step identifies skin regions in the input image by comparing color values of each pixel to a human skin color model. Considering the processing time limit of a real application, in the algorithm we use YUV color model instead of more complex model employing Farnsworth nonlinear transformation. Skin area detection is still based on pixel operations.

♦ *Contour following*:

As an image segmentation process, contour following uses a 3x3 filter to extract boundary of each object region obtained by previous processing steps. In the next step, the boundaries are used to calculate super-ellipse parameters.

♦ *Super-ellipse fitting:*

Super-ellipses are extensions of ellipses. Super-ellipse fitting uses Levenberg-Marquardt minimization method [7] to find the best parameter set containing five parameters for each region.

♦ *Graph matching:*

In the result of super-ellipse fitting, the regions in the input image are represented by small sets of parameters. In graph matching step, unary features of a region and binary feathers between regions are extracted and are compared to the reference graphics in a database. Graph matching uses techniques of pattern recognition. In this way, the algorithm can recognize the body parts of human beings and their poses.

The basic architecture of the program is shown in Figure 2. Major blocks of the program are functions *RegionExtraction*, *ContourExtraction*, *SuperFit*, and *Match*. *RegionExtraction* performs background elimination and skin area detection. *ContourExtraction* calls *Contour* subroutine to extract the boundary of all the regions. *SuperFit* is the block of super-ellipse fitting for all the boundaries extracted by *ContourExtraction*. In the *SuperFit* block, *super3* is called to perform super-ellipse fitting for the boundary of one region. *Match* first calls *adjacent* to obtain the adjacency information among regions, then it performs graph matching to identify human body parts. The four block of the algorithm corresponds to four different fields in image processing—color image processing, image segmentation, image description and pattern recognition [6]. In the following sections, we will use *region*, *contour*, *super* and *match* to represent the major blocks *RegionExtraction*, *ContourExtraction*, *SuperFit* and *Match* correspondingly.

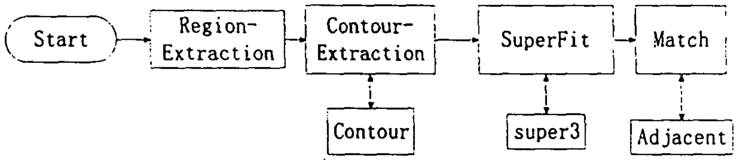


Figure 2: Architecture of the target program

3 Evaluation Environment

This section introduces the two major evaluation tools used in the experiments—SimpleScalar tool set and TriMedia SDK.

3.1 SimpleScalar

SimpleScalar tool set is a suite of public domain simulation tools. The superscalar architecture of SimpleScalar is derived from the MIPS-IV ISA. The tool set takes compiled binaries and simulates their execution. A modified version of GNU GCC compiler is provided along with SimpleScalar tool set. Therefore, programs coded in C language can be simulated on SimpleScalar simulators.

In our experiments, we use three simulators —sim-cheetah, sim-profile and sim-outorder running on an Ultra-Enterprise-4000 with Solaris 5.7 operating system.

Sim-profile is a profiling simulation tool. Using a functional simulating engine, sim-profile can generate detailed profile on instruction classes, branches, memory access modes, and data segments.

Sim-cheetah is a functional simulator for cache behavior analysis. Miss ratios can be collec'ed under different cache parameters such as level (one level/two level), size, type

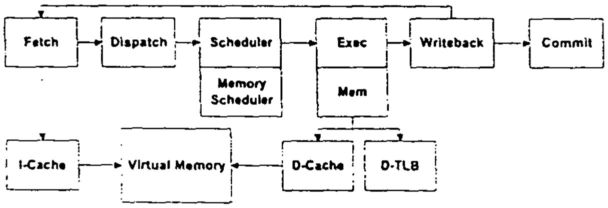


Figure 3: Pipeline for sim-outorder

(data/instruction/unified), associativity, and line size. In the experiments, sim-cheetah is used to analyze the impact of different cache size, associativity, and cache line size.

Different from previous two functional simulators, sim-outorder is a detailed architectural simulator. The architecture of the sim-outorder is shown in Figure 3. It provides information that is more accurate by simulating out of order issue and execution using reorder buffer. As a superscalar processor, sim-outorder is employed to compare with a VLIW (Very Long Instruction Word) processor TriMedia TM1300 processor in our experiment.

3.2 TriMedia

Designed for media processing, TriMedia processing board allows Windows and Macintosh platforms to take advantage of the TriMedia processor via PCI interface. Multiple TriMedia processing board can be install to one host PC to provide multiprocessing ability. A TriMedia board has a TM1300 TriMedia processor with its own dedicated memory. A 32-bit TM1300 TriMedia processor has a five issue VLIW (Very Long Instruction Word) CPU together with several coprocessors as shown in Figure 4. The CPU in the processor has multiple functional units and 128 registers. Table 1 shows the major features of a TriMedia CPU.

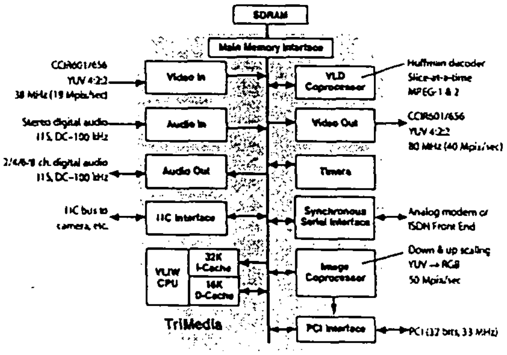


Figure 4: Structure of a TriMedia processor

#Functional Unit	Constant	5
	Integer ALU	5
	Load/Store	2
	DSP ALU	2
	DSPMUL	2
	Shifter	2
	Branch	3
	Int/Float MUL	2
	Float ALU	2
	Float Compare	1
	Float sqrt/div	1
#Register		128
Instruction cache		32KB, 8 way
Data cache		16KB, 8 way
#Operation slots/instruction		5

Table 1: Features of TriMedia

Besides its complicated hardware, TriMedia board comes with a set of powerful software tools, which includes tmsim simulator providing full functional simulation. During the experiment, we use the TriMedia Software Develop Kit version tcs2.20 that includes a compiler tmcc, an assembler tmas, a linker tmld, a simulator tmsim, an execution tool tmrun, and a simulator tmprof. The TriMedia system is running in a Dell Precision-210 computer with two TriMedia reference boards.

While SimpleScalar tool set is strong at providing profile information and performing cache behavior analysis, TriMedia system can provide runtime information for VLIW architecture and multiprocessing. Start from next section, we will present experiments using these tool sets.

4 Workload characterization

Most of the workload characterization results are collected by using SimpleScalar. This section presents workload characteristics of the target algorithm including instruction frequencies, branch class frequencies, and memory access pattern.

There are several steps in our algorithm. However, SimpleScalar tool set does not offer profile based on function. Therefore, the whole algorithm is divided into several separate programs. The programs exchange data with each other via files. While this method can get information for individual step, it introduces extra overhead for each step. To solve this, we run the file processing part of each program separately and subtract the overhead. Profiles based on algorithm blocks are collected using this method.

4.1 Instruction frequencies

Keep balanced resource distribution in a video processor is of critical importance. In order to achieve the proper balance, one needs to consider the instruction frequencies. In this part of work, we used SimpleScalar profiling program sim-profile to gather this information. The results are shown in Figure 5. With these data, the resource allocation among functional units can be calculated. For a CPU is used to process the whole algorithm, the resource allocation is (3:1:5:1) for load/store units, branch units, integer ALUs, and floating point units respectively.

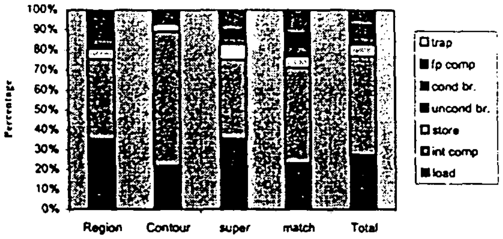


Figure 5: Instruction class profiling

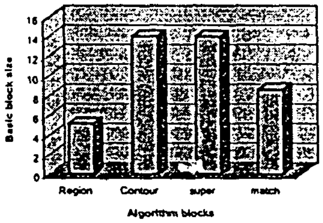


Figure 6: Basic block size

4.2 Basic block and branch statistics

Basic block is a block of instruction code that does not contain branch instructions. The average size of basic blocks in a program reflects its instruction level parallelism. Using instruction class profiling, we can calculate the average basic block size as

$$average\ basic\ block\ size = \frac{\#\ total\ instructions}{\#\ branch\ instructions} = \frac{1}{percentage\ of\ \#\ branch\ instructions}.$$

The results lead to Figure 6. While *region* algorithm block has a relative small average basic block size of five instructions/block, other algorithm blocks have average basic block sizes near or over ten, which suggest large amount of instruction level parallelism in the algorithm.

In addition to basic block size, the frequencies of different branch instruction classes can contribute to efficient hardware and compiler design. The branch instruction class profiling also is performed by sim-profile. The results are shown in Figure 7. Since the conditional direct branches are a major part of branch instructions in all the algorithm blocks, we expect that a successful branch prediction

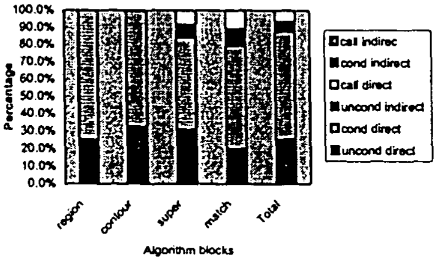


Figure 7: Branch operation frequencies

scheme would result a considerable increase of the instruction parallelism. Moreover, the results indicate that a considerable percent of branch instructions are unconditional direct branch. This fact implies that unrolling or grafting scheme can be used to speed up the programs.

4.3 Memory accessing pattern

A significant part of a modern processor design is the design of the memory interface. Correspondingly, memory access pattern of a program plays an important role in workload characterization. In our work, we used SimpleScalar again to collect the information about memory access of the target programs.

4.3.1 Cache behavior

Cache is of paramount importance in modern processor. In this part of the work, sim-cheetah is used to analyze the cache behavior of the algorithm. For the reason that cache behavior is exhibited by a program as a whole, the entire algorithm instead of separated parts is used in the evaluation. In following parts of this subsection, we present the evaluation of working set size of the target algorithm and the effect of different cache associativities and cache line sizes.

Working set size is a measurement of cache size needed by a program. It is identified as the cache size where the miss ratio decreases dramatically with respect to smaller cache size. In case that there is no dramatic miss decrease, working set size is the smallest size, which gives a miss ratio below 3%. During the experiments, direct-mapped cache was evaluated for all base 2 sizes between 1 KB and 1MB, using a line size of 64bytes. Three classes of cache, data cache, instruction cache, and unified cache, are tested. The results are shown in Figure 8. For an instruction cache, a size of 1KB is enough to reduce the miss ratio to 2% corresponding to the relatively small size of the instruction codes. Although the size of the data processed by the program is large, the program does not require a large data. Only 8KB cache is required to reduce the data cache misses below 3%. In addition, unified cache is quite fit for the application with a working set size at 4KB.

In addition to working set size, an important parameter for cache is associativity, which is the number of potential cache blocks associated to a particular memory address. Figure 9 shows the effects of different cache associativities for a 1KB cache, using a line size of 64 bytes. The figure suggests that cache associativity has significant impact on cache performance in that a two-way 1KB data cache outperforms a direct-mapped 8KB data cache. If we consider the impact of large associativity on latency of cache access, an associativity of 2 or 4 is recommended for the target application.

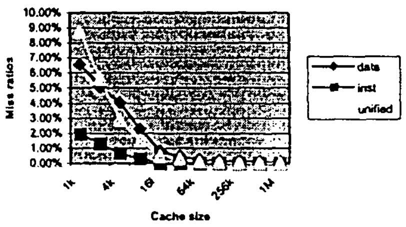


Figure 8: Overall cache miss ratios

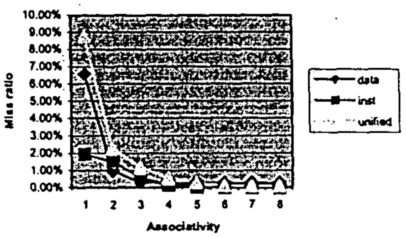


Figure 9: Cache miss ratios of different associativities

Another major parameter of a cache is its line size. As video applications usually require large amount of data, one would expect larger line size would bring better performance. However, our experiment results shown in Figure 10 and Figure 11 indicate this is not always the case. In experiment, a cache with 4KB size is used. For an instruction cache of both associativities, a line size of 128 bytes or 256 bytes would be optimal. However, for data and unified cache, a line size at 32 bytes or 64 bytes can bring lowest miss ratios. The Larger cache line size would significantly degrade the performance.

Put all the above results together, recommended cache settings would be 2 to 4 way unified 1KB cache with a line size at 64 bytes.

4.3.2 Address mode

Address mode is the way that an address of a memory access is formed. It affects the data path design of a processor. During the experiment, we use sim-profile to collect this information. The results are shown in Figure 12, where *reg* marks registers, *fp* represents far pointers, *sp* is for small pointers, *gp* stands for global pointers, and *const* represents constants. All the programs make intensive use of *fp+const* address mode. *Contour* block is different from other block in term of address mode since it uses a considerable percentage of *const* address modes. Since all the address mode except *(reg+reg)* mode are used in some parts of the algorithm with a considerable percent, the CPU for smart camera should support these address mode except *(reg+reg)* mode in its architecture.

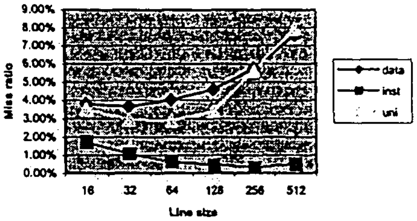


Figure 10: Cache miss ratio of different cache line sizes (directly mapped cache)

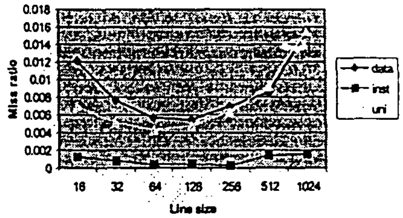


Figure 11: Cache miss ratios of different cache line sizes (two way caches)

4.3.3 Segment access pattern

For a program, there are usually four segments—text, heap, stack, and data. Instructions are normally stored in text segment and as its name suggests, data segment hold static data. Heap and stack are associated with dynamic data. Stack provides spaces for auto type variables as those

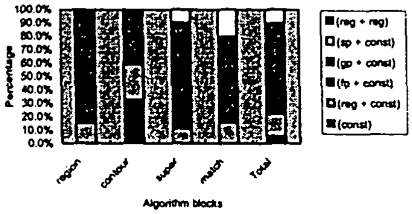


Figure 12: Address mode profiling

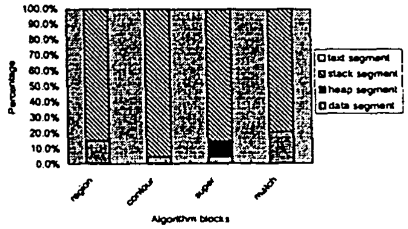


Figure 13: Data access segment profiling

declared in a function and is responsible for the function parameters and the returned variables. Heap is the place where the dynamic allocation occurs. The information about the segment access pattern can help us to understand the memory usage of an algorithm and help compiler to balance the memory space between different segments. Figure 13 shows the experimental results collected by sim-profile. The stack segment is obviously the most heavily used segment. Another information offered by the figure is that *Super* has accesses to heap. Since this is a real-time application, it is possible that we reduce the overhead of allocating and release memory block by using global data.

4.4 Comparison between VLIW processor and superscalar processor

VLIW Architecture represented by TriMedia TM1xxx processor, TMS320C6X etc and superscalar architecture used in the Intel Pentium, the Ultra Sparc etc, are two major architectures for modern processors. Both kinds of processors take advantage of instruction level parallelism by using multiple functional units. While superscalar processors exploit instruction level parallelism by using hardware, VLIW processor relies on advance compiler techniques to simplify hardware. For video processing applications, the large amount of instruction level parallelism is evident. A comparison between the performances of two processors with different architectures is helpful to choose a better architecture for smart camera system.

In the experiment, we use SimpleScalar simulator sim-outorder and TriMedia simulator tmsim. The configurations of the two simulators are shown in Table 3. To be fair, the program tested is not optimized by hand.

Configuration	SimpleScalar1	TriMedia
#Integer ALU	4	5
#Float ALU	2	2
#Register	32	128
Window size	16	-
# MUL/Div	4	5
#Mem port	2	2
Compiler	Gcc v 2.6.3	Tmcc v 5.7.1
Compiler optimize option	-O3	-O3

Table 3: Configuration of the simulators

The results are in Table 2. Since one instruction in a VLIW processor may contain several operations, we use "Operation" in the table. The results in table indicate that superscalar architecture is better than the VLIW processor. However, since VLIW architecture has a simpler hardware implementation, its clock cycle would be shorter than that of superscalar architecture. Considering this, we would expect close performance for both architectures. Moreover, as we will show in next section, manual optimization can significantly reduce the running time of the program on a TriMedia processor. It is realistic to manual optimize a program in an application-specific system. Thus, we recommend use VLIW architecture in smart camera system.

	SimpleScalar	TriMedia
Operations	3.40e+07	5.99e+07
Execution Cycles	1.84e+07	2.78e+07
Operations per cycle	1.85	2.15

Table 2: Comparison between superscalar and VLIW processor

5 Optimization of the proposed algorithm

While optimization can speed up target program, which helps to the design of the smart camera system, it also helps to identify effective ISA (Instruction Set Architecture) features, which can be used in new microprocessors. During the optimization, we explored algorithm level optimization relying on the understanding of the algorithm as well as low-level optimization method such as loop unrolling, unrestricted pointer, and custom operations. Microsoft Visual C++ 6.0 professional and TriMedia tool sets are used to evaluate the effectiveness of the optimization. In the experiment, Visual C++ is running on an IBM ThinkPad i1400 with windows 98 operating system. While Visual C++ can offer suitable function support for algorithm level optimization, two factors make the running time collected by Visual C++ varies slightly from one experiment to another. One is that Windows 98 is a multitask operation system, so the other programs running simultaneously may affect the total running time of the target program. Second is that the cache may effect the running time. However, algorithm level optimization can bring significant change to the running time so that the variance does not interfere with the observation of the optimization effects.

5.1 Algorithm level optimization

Algorithm level optimization replaces time-consuming code segment by code that is more efficient. It helps to deep understand of the target program.

It is evident the optimization should be performed on the most time consuming parts of the algorithm. To make this, a critical step is to collect time information of each function. While the TriMedia tools can offer running time of each function without the time of its sub-functions, Microsoft Visual C++ profiler can collect running time of each function both with and without the time of its sub-functions. The later is more suitable for high-level optimizations since reducing the number of calls to its sub-function could reduce the total running time of a function.

Function	Execution Time (Milliseconds)	Percentage
RegionExtract	7.029	16.1%
ContourExtraction	6.668	15.3%
SuperFit	29.881	68.6%
Match	7.388	17.0%
Total	50.966	100%

Table 4: function execution times of original program

The original time distribution for major functions in the algorithm is shown in Table 4. The super-ellipse fitting step is the most time consuming part. Therefore, we consider modifying the super-ellipse fitting. By using moment-based initialization to replace the original method developed from Principle Component Analysis, we can remove Levenberg-Marquardt fitting procedure and cut down the execution time of the super-ellipse fitting. The results are shown in Table 5.

Function	Execution Time (millisecond)	Percentage
RegionExtract	6.915	49.0%
ContourExtraction	7.065	50.1%
SuperFit	0.129	0.9%
Match	7.44	52.7%
Total	21.549	100%
Adjacent	7.262	51.5%

Table 5: Function timing profiling after optimization on super-ellipse fitting

Function	Execution Time (millisecond)	Percentage
RegionExtract	6. 954	50. 7%
ContourExtraction	6. 644	48. 5%
SuperFit	0. 111	0. 8%
Match	0. 710	5. 2%
Total	14. 419	100%
Adjacent	0. 613	4. 5%

Table 6: Running time of the functions after optimization on *Adjacent*

Now that graph matching is the primary part. Further data indicate that adjacent is the major time consuming part of the function. *Adjacent* is a function that determines the adjacency of two super-ellipses using pixel information. A new idea is that parameter information may be used to determine the adjacency. The effectiveness of the change is evident—the time of the *adjacent* function slashes as shown in Table 6.

After these optimizations, the total execution time of the algorithm dropped from 51 milliseconds to 14 milliseconds. Now, we consider the low-level optimization skill.

5.2 Low level optimization for TriMedia processor

Five issue VLIW TriMedia processors offer abundant resources for instruction level parallelism. While the difference between the average basic block size of the algorithm and the instructions per cycle of the program suggest that manual optimization may reduce the running time dramatically.

On a TriMedia processor, the processing times for the 10 most time consuming functions and the total running time are listed in Table 8 where functions RegionExtract, Contour, and ContourExtraction are the most time consuming parts.

Function	Total Cycles (x1000)	(%)
RegionExtract	2737	36.66%
Contour	2054	27.51%
ContourExtraction	1091	14.61%
abs	369	4.94%
memcpy	262	3.5%
adjacent	177	2.37%
GetNext	107	1.44%
sin	94	1.26%
cos	86	1.15%
memcpy	85	1.14%
Total	7470	100%

Table 8: Function timing before Optimization

Function	Total Cycles (x1000)	(%)
RegionExtract	2493	38.04%
Contour	2052	31.32%
ContourExtraction	425	6.48%
abs	369	5.63%
memcpy	262	3.99%
adjacent	177	2.7%
GetNext	108	1.64%
sin	94	1.43%
cos	86	1.31%
memcpy	85	1.3%
Total	6550	100%

Table 9: results for using restrict pointers

The first step is to use restricted pointer (A restrict pointer is such a pointer that that no other variable, pointers will alias the objects refer by restrict pointer for as long as the pointer is in scope.[5]). The results are shown in Table 9. A considerable running time reduction is achieved especially for *ContourExtraction*, whose execution cycles are reduced by more than half. The total number of execution cycles drops by more than 10%.

Then, using loop unrolling and applying the custom functions offered by TriMedia, we eliminate the most part of branches in region extraction. In addition, we simplified the condition test in Contour. The impact on running time is significant as shown in Table 7. In this step, total execution

Function	Total Cycles (x1000)	(%)
Contour	1518	34.86%
RegionExtract	1168	26.81%
ContourExtraction	552	12.68%
memcpy	254	5.84%
adjacent	177	4.06%
sin	93	2.14%
memcpy	88	2.02%
cos	85	1.95%
memset	62	1.43%
fmod	49	1.11%
Total	4356	100%

Table 7: Function timing after Optimization

time is reduced by 33%. The custom function used in this optimization is INONZERO, the results strongly suggestion that a media processor should include such instructions for replacing branches.

6 Multiprocessor architecture for smart cameras

Video processing for smart camera requires intensive computation, so it may be necessary to multiple processors for enough processing ability. In such a case, the organization of the processors rises as an important issue. In order to distinguish it from micro-architecture, we call this architecture macro-architecture. In this section, we will discuss two macro-architectures for smart camera processing unit.

6.1 Parallelism in video processing and its impact on macro-architecture

For our video-processing algorithm, there are several different levels of parallelism. One is instruction level parallelism we have analyzed in previous sections. This parallelism is in conjunction to the symmetry of the pixels in an image. The VLIW architecture is corresponding to this level of parallelism. Another one is the parallelism between different processing phases of the algorithm. In our algorithm, the different phases of processing can be performed simultaneously if they are processing different frames. The corresponding parallel architecture is the macro-pipeline architecture as shown in Figure 14. The third one is the parallelism in conjunction to the symmetry of the frames in the video stream. Although pose recognition need the sequential information of the frames, the front processing steps such as skin color detection, contour following, super-ellipse fitting, etc do not require the sequential information. Thus, there exists an inter-frame parallelism for this video-processing algorithm. The symmetric parallel architecture shown in Figure 15 is proposed for this level of parallelism. While the VLIW is microprocessor architecture, the later two architectures are above micro-architecture level, we call them macro-architectures.

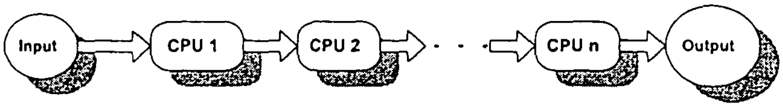


Figure 14: Macro-pipeline Architecture

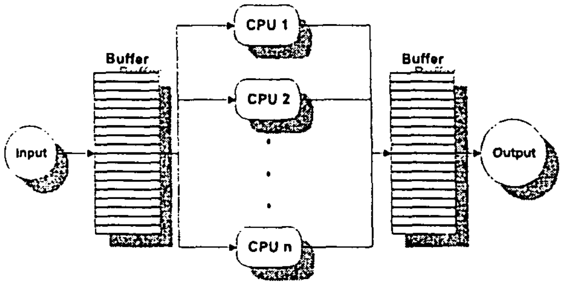


Figure 15: Symmetric Parallel Architecture

In our work, a set of experiments is conducted to compare the efficiency of different macro-architectures.

6.2 Cost of Communication

Apart from computational issues discussed in previous sections, communication is of importance in multi-processor systems. There are two major catalog of communication operation--- one is data transference and the other is synchronization. Cost of data transfer between TriMedia boards is evaluated using two TriMedia reference boards in a host PC. The TriMedia boards use shared memory to transfer data. There are two shared integer flag fg1 and fg2 in the shared memory space. Fg1 is initiated to 0 and fg2 to 1. Two program prog1 and prog2 to are running in two TriMedia boards. Program prog1 waits for fg2 turning to 1. If fg2 is 1, prog1 sets it to 0 and copies a frame of data (384x240 bytes) to shared memory space and then sets fg1 to 1. Program prog2 in another TriMedia boards waits for fg1 turns to 1. When fg1 turns to 1, prog2 sets it back to 0 and copies the frame in the shared memory space to its own data space. After doing this, prog2 sets fg2 to 1. Then another recurrence begins. In our experiment, 10,000 such loops take 25 seconds. Thus, it takes 2.5 milliseconds to transfer one frame of data from one TriMedia Board to anther.

Cost of synchronization is tested on an IBM ThinkPad i1400 (Intel Celeron 500, 128MB memory) with window 98 operating system. Semaphores are used for synchronizing. A semaphore is a system maintained variable. The read/write operations on a semaphore are atomic operations. The so-called atomic operation is defined as an operation that no other operations on the same variable can start before the operation finishes. In the experiment, two semaphores s1 and s2 are created and used. Both semaphores have the max value of 1. S1 is initiated to 0 and s2 to 1. Two process p1 and p2 are running in the experiment. The process p1 requests s2 from the operation system. When s2 is available (s2 is greater than zero), the system decreases s2 by one and wakes up process p1 if it is pending. Then p1 releases s1 to the operating system (increasing s1 by one). Process p2 waits for s1 to be available. When p2 obtains s1, p2 releases s2 to the operating system. This is the process of one loop. Since 100,000 loops take 3,840 milliseconds in the experiment and each loop has four semaphore operations, one semaphore operation takes about 9.6 *us*.

6.3 Feature of algorithm for macro-architecture

For the purpose of analysis macro-architecture, some features other than workload characteristics in micro-architecture are needed. We evaluate these features and list them in Table 10. The data are results from SimpleScalar simulator by using un-optimized programs. It is shown in Table 10 that *super* and *match* have relatively high standard deviation. For video input, if a frame needs a long processing time, we would expect the frames near it need long processing time too. Thus, when designing the video processing units for a smart camera, one needs to consider the worst case instead of average case. For this reason, we will use the worst-case data in the following analysis. Input and output data sizes for each algorithm block are fixed by program and are listed in the table.

		Region	Contour	Super	Match	Total
#Operation Executed (one frame)	Average	6.81E+06	1.69E+07	2.08E+07	2.64E+07	7.09E+07
	Percentage (Average)	9.6%	23.8%	29.4%	37.2%	100%
	Standard Deviation	7.37E+04	2.15E+05	5.61E+06	2.33E+07	2.86E+07
	Max	6.92E+06	1.72E+07	2.75E+07	6.68E+07	1.18E+08
	Percentage (Max)	5.9%	14.6%	23.3%	56.6%	100%
	Min	6.73E+06	1.66E+07	1.21E+07	4.53E+06	4.00E+07
Input data size (bytes)		430080	107520	2.20E+04	4804	430080
Output data size (bytes)		107520	22048	4804	-	-

Table 10: Algorithm feature for macro-architecture analysis

6.4 Comparison between two macro-architecture

With all data in the previous subsection ready, a comparison between two macro-architectures is made. In the comparison, we use a two-CPU setting for both architectures. The maximum potential speedup is the speedup provided with unlimited number of CPUs. For a pipelined architecture, the maximum potential speedup can be calculated as $max\_speedup = \frac{1}{\max(percentage\ of\ processing\ time)} = 1.77$ where percentage of processing time is the percentage of each algorithm block's processing time in the total processing time. Suppose a CPU with 100MHz clock rate can execute 2 operations per cycle, it takes 0.59 second to process one frame. If we have an input rate at 10 frame/second (This is the rate can be processed by human being's visual system.), the max speedup for symmetric parallel architecture is

$$max\_speedup = one\_frame\_processing\_time \cdot input\_rate = 5.9\ .$$

For pipeline architecture, $throughput = \frac{1}{\max(average\ processing\ time\ on\ a\ CPU)}$.

For symmetric architecture, $throughput = \frac{\#CPU}{average\ processing\ time}$.

Latency is defined as the period from the time a frame starts being processed to the end of the processing. The latency for pipeline architecture is calculated as

$$\begin{aligned} latency &= \#pipeline\_stages \cdot processing\_time\_per\_stage \\ &= \#pipeline\_stages \cdot max(average\ processing\ time\ on\ a\ CPU) \end{aligned}$$

The latency for a symmetric parallel architecture is $latency = average\_processing\_time$.

The results listed in Table 11 show that the macro-pipeline architecture requires a smaller memory, while symmetric parallel architecture offers a better speedup, scalability and needs less communication.

Macro-architecture	Macro-pipeline	Symmetric parallel
Max potential speedup	1.77	5.9
Memory (bytes)	586500	1075200
Speedup	1.77	2
Through put (frames/sec)	5.0	5.6
Communication data size (bytes)	4.57E+05	3.22E+05
Latency (sec)	0.801	0.709
# Minimum Synchronization operation	4	2
Special require	Pipeline structure of algorithm	Processing does not require sequential information
Scalability (Change program for different #CPUs)	Yes	No

Table 11: Comparison between Macro-pipeline and symmetric parallel architecture

7 Conclusion

This paper presents a workload evaluation for a smart camera system. Taking advantage of SimpleScalar tools set and TriMedia SDK tools, we analyze application-driven architecture tradeoffs.

In the experiments, we examine a variety of micro-architecture level characteristics including operation frequencies, basic block sizes, branch class frequencies, data address modes, working set sizes, cache associativities, cache line size, data segments accesses and superscalar/VLIW tradeoffs. The instruction frequencies analysis recommends a resource allocation ratio of (3:1:5:1) for load/store units, branch units, integer ALUs, and floating point ALUs. Cache parameter evaluation recommends a 2 to 4 way unified 1KB cache with a line size at 64 bytes for each processor.

In addition, manual optimizations are performed in algorithm level to explore the high level features as well as in instruction level to explore the ISA features of TriMedia processor. Algorithm level optimizations yield a three times speedup by applying additional information. Low-level optimization offers another two times speedup by using restricted pointers, custom operation and loop unrolling. The effectiveness of the custom operation INONZERO suggests that ISA of the smart camera system incorporating such conditional instructions.

Moreover, two macro-architectures are evaluated. While macro-pipeline architecture required smaller memory, symmetric parallel architecture can provide better a speedup and scalability.

Further work will involve cooperation of processors in different smart cameras and evaluation of the real time performance for the proposed macro-architectures.

Reference

[1].B. Ozer, W. Wolf, "Video Analysis for Smart Rooms", SPIE, ITCOM 2001, Denver, CO, August 2001

[2].B. Ozer, W. Wolf, A. N. Akansu, "Relational Graph Matching for Human Detection and Posture Recognition", SPIE, Photonic East 2000, Internet Multimedia Management Systems, Boston, MA, November 2000

[3].J. Fritts, W. Wolf, and B. Liu, "Understanding multimedia application characteristics for designing programmable media processors," SPIE Photonics West, Media Processors '99, San Jose, CA, January 1999, pp. 2-13.

[4].D. Burger, T. M. Austin. "The SimpleScalar tool set, version 2.0", In Technical Report 1342, University of Wisconsin Madison, CS Department, June 1997

[5].TriMedia Technologies Inc., " TriMedia Documents", http://www.trimedia.com/products.html#download, October 2000.

[6].R. C. Gonzalez, R. E. Woods, "Digital image processing", Addison-Wesley Publishing Company, Inc

[7].W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, "Numerical Recipes in C", Cambridge University Press, Second Edition, 1995

Human Activity Detection in MPEG Sequences

Burak Ozer[†] Wayne Wolf[‡] Ali N. Akansu[†]

[†]Department of Electrical and Computer Engineering
New Jersey Institute of Technology
New Jersey Center for Multimedia Research
Newark, NJ 07102, USA
ibo8175@oak.njit.edu

[‡]Department of Electrical Engineering
Princeton University
New Jersey Center for Multimedia Research
Princeton, NJ 08540, USA
wolf@ee.princeton.edu

Abstract

In this paper, we propose a hierarchical method for human detection and activity recognition in MPEG sequences. The algorithm consists of three stages at different resolution levels. The first step is based on the principal component analysis of MPEG motion vectors of macroblocks grouped according to velocity, distance and human body proportions. This step reduces the complexity and amount of processing data. The DC DCT components of luminance and chrominance are the input for the second step, to be matched to activity templates and a human skin template. A more detailed analysis of the uncompressed regions extracted in previous steps is done at the last step via model-based segmentation and graph matching. This hierarchical scheme enables working at different levels, from low complexity to low false rates. It is important and interesting to realize that significant information can be obtained from the compressed domain in order to connect to high level semantics.

1 Introduction

Most human activity recognition techniques are implemented in the uncompressed domain and depend on proper segmentation of the human body. The purpose of our work is to investigate activity recognition in the compressed domain in order to reduce computational complexity and avoid dependency on correct segmentation in an uncompressed image. The algorithm consists of three stages at different resolution levels. The first step is based on the principal component analysis of MPEG motion vectors to match the detected activity with known human activities; namely, walking, running, and kicking. The motion vectors are grouped automatically according to velocity, distance and human body proportions. The algorithm uses DC-DCT coefficients of the luminance and chrominance values when more detailed information is needed. These values are matched to activity templates and a human skin template. The finest details in the sequences are obtained from the uncompressed domain via model based segmentation and graph matching. This hierarchical scheme enables working at different levels, from low complexity to low false rates. Our proposed graph-based representation [1] enables the detection of human presence in the frames, as well as posture recognition by using the DC-DCT coefficients in the compressed and pixel values in the uncompressed domains. The major contribution of the graph matching method is the automatic creation of semantic segments from the combination of low-level edge or region-based segments using model-based segmentation. The generality of the reference model attributes allows the detection of different postures while the conditional rule generation decreases the rate of false alarms.

Since image and video applications are generally represented in the compressed domain, such as JPEG or MPEG, there is a need for image/video manipulation and automatic content extraction in the compressed domain. As stated in [2], for existing compression standards the compressed-domain image/video manipulation techniques can be used to help solve the bandwidth problem. Hence applications without expanding the coded visual content back to the large uncompressed domain would reduce the need of large bandwidth and intensive computing. The use of available information in compressed video and images mostly has been investigated for video indexing, and shot and scene classification. In [3], hierarchical decomposition of a complex video is obtained using scaled DC coefficients in an intra coded DCT compressed video for browsing purposes. The technique combines visual and temporal information to capture the important relations within a scene and between scenes in a video. A general model of a hierarchical scene transition graph is applied for video browsing. In [4], the authors examine the direct reconstruction of DC coefficients from motion compensated P-frames and B-frames

0-7695-0939-8/00 $10.00 © 2000 IEEE

of MPEG compressed video. Their analysis and experimental results show that lower cost approximations can be used successfully for various image processing operations, such as shot detection, shot matching and clustering. In [5], an automatic scene classification scheme is proposed for MPEG videos. The scenes are divided into low, medium, and high texture and activity scenes. The bit rates of the I, P and B frames are used in shot texture classification while the percentage of macroblock types are used for shot motion classification. In [6], a metric based on the mean and the variance of MPEG motion vectors is used to classify the scenes according to the activity level.

MPEG motion vectors are used mostly to index videos (low-high activity) and track objects. The object detection in the compressed domain is more restricted since this application requires more detailed information. In [7], an object tracking algorithm is proposed using compressed video only with periodically decoding I-frames. The object to be tracked is initially detected by an accurate but computationally expensive object detector applied to decoded I-frames. In [8], an algorithm to detect human face regions from dequantized DCT coefficients of MPEG video is proposed. The algorithm uses the DC DCT values of chrominance, shape, and energy distributions of the face area. This method is suitable for color images with face regions greater than 48 by 48 pixels (3 by 3 MPEG macroblocks). The authors extend their work in [9] in order to track and summarize faces from compressed video. The previous algorithm is used to detect faces and MPEG motion information is used with the Kalman filter prediction to track faces within each shot. The representative frames are then decoded for pixel domain analysis and browsing.

The data sets for human detection applications in the compressed domain include anchorperson scenes, news stories and interviews, where the faces and the upper-bodies occupy large areas in the image. However, at lower resolution, available motion vectors can be used to detect human activity by comparing it with known human activity patterns. Our work can be divided into three major parts. The first part is activity recognition in the compressed domain based on principal component analysis (PCA) [10]. In Figure 1, the MPEG motion vectors and motion vector groups according to the human body proportion, are displayed. In the second part, DC DCT differences between frames in the compressed domain are matched to activity templates (side-view), obtained from a training set, to distinguish activity periods. The DC coefficients are also used in the graph matching algorithm for human body recognition in the compressed domain, but this method is suitable for images with face regions greater than 3 by 3 macroblocks. Since graph matching performance depends highly on face detection, this is a crucial restriction. In most cases, the resolution of the face area does not satisfy this criteria, which leads us to implement the graph matching algorithm in the uncompressed domain for the finest analysis of the human body and posture. This paper is organized as follows: Processing in the compressed domain is given in the next section where activity recognition based on principal component analysis is explained. The third section covers the template and graph matching algorithms at higher resolutions. The results are displayed in section 4.

Figure 1. Motion vectors and vector groups.

2 Activity Recognition Using MPEG Motion Vectors

Activity recognition problem can be divided into two subparts: the first one is collecting satisfactory measurements and the second one is developing a recognition algorithm based on these measurements. Most of the related work use activity measurements from uncompressed images after a proper segmentation of human body parts. Our measurements are obtained from MPEG motion vectors of macroblocks. Since the resolution of the motion vectors is one macroblock and there is no direct correspondence with the object parts and their motion, a robust and global model must be used. The corruption of data is another problem in MPEG motion vectors since some blocks can not be tracked during some frames. An overview of research on human motion analysis can be found in [11], [12]. The major problems in the activity recognition is the scale, shift and projection changes between the model and the test data and segmentation dependency. One of the activity modeling methods proposed in [13] is based on first order Markov model descriptions and continuous propagation of observation density distributions. Hidden Markov Models are used to predict the state transitions. In [14], speed and direction components of 2D trajectories are represented by scale-space images that are invariant to Euclidean transforms. A method based on time-frequency analysis is proposed in [15] to detect periodic human motion with self-similar characteristic. The outline of the human body is used to detect the periodical relative limb movement in [16] by a template matching process. In these approaches, for each activity, a separate model is developed in order to compare with the observed activity. These approaches are robust to local transformations but lack a global detailed model to capture the variabilities. Principal component analysis method is one of the global approaches. Our activity recognition model is based on the principal component analysis which has also been used by Yacoob and Black [10] for human activity recognition in uncompressed video sequences. The authors use the motion measurements for segmented human body parts. In our method, we first detect the moving regions and then group the motion vectors automatically by using the ratio of the human body parts. Hence the measurements do not correspond to the actual human body parts but to macroblock groups corresponding to human region. For the classification of moving regions, the neighboring blocks with a velocity greater than a predefined threshold are classified as one moving object. The following subsection covers the principal component analysis.

2.1 Principal Component Analysis

• **Step1:** PCA was successfully used for face recognition. A compact representation of facial appearance is described in [17], where face images are decomposed into weighted sums of basis images using a Karhunen-Loeve expansion. The eigenpicture representation has been used in [18] as eigenfaces for face recognition. PCA is a dimensionality reducing technique, used in pattern recognition. It reduces dimensionality by projecting the motion vectors to a new space spanned by the training data set. For training the system, several walking, running and kicking man sequences which are temporally aligned are used. For these sequences, the object region is extracted by grouping MPEG motion vectors. Then, the object is segmented to three parts (upper body, torso and lower body) according to the human body proportions. The mean of the motion vectors in horizontal and vertical direction is computed for the macroblocks corresponding to each part (6 parameters) for a number of sequences T. A training set of k different examples for each activity forms matrix A of dimensions $6T \times k$. Then the singular value decomposition of the matrix A is computed to get the approximated projection of the exemplar vectors (columns of A) onto the subspace spanned by the $q < k$ basis vectors. Hence activity basis with parameters m are computed.

$$A = U\Sigma V^T \tag{1}$$

where A is the motion parameter matrix, U represents the principal component directions, Σ includes the singular values, and V^T expands A in principal component directions. To recognize the activities, an unknown sequence, other than test sequences of an activity which can be shifted and scaled in time is compared with the training set. The normalized distance between the coefficients c_i from the observed data set and coefficients of exemplar activities m_i is used to recognize the observed activity that is transformed by the temporal translation, scaling and speedup parameters [10]. The Euclidean distance is given as

$$d^2 = \sum_1^q (c_i/||\mathbf{c}|| - m_i/||\mathbf{m}||)^2 \tag{2}$$

Figure 2. Frames from walking, running, and kicking man training sets.

The algorithm is applied for recognition of three activity classes: walking, running, and kicking. 10 training test sequences for each class are obtained from various sources for the side-view. The camera motion is assumed to be zero. In Figure 2, some test frames from the activity training sets are given. The detection of the moving regions and the determination of the activities from the grouped MPEG motion vectors gives a coarse information about the scene.

3 Human Detection and Posture Recognition

The second step of the algorithm uses 8 by 8 block information (DC values) in the frames where human activity has been detected from the motion information. The last step is based on graph matching where nodes correspond to the human body parts in the uncompressed domain. This step is also implemented in the compressed domain where the graph nodes correspond to the segments of DC DCT values.

• **Step2:** The difference of the DC values for 8 by 8 blocks between consecutive frames are computed and the difference image is binarized by thresholding. To train our system, we used several human activity sequences from side-view with the similar camera distance, human motion direction and velocity. In order to find the template for each body position during one activity period, the mean of the moving regions, corresponding to these positions, are calculated. The classification is done by using a basic template matching measure. Note that the mirror image of the template is also used. For every DC-DCT difference frame, the blocks are compared to the activity templates. For scale change invariance, we scaled the moving block regions with different scale parameters and calculated the matching value for each scale factor.

• **Step3:** Note that the sequences of interest include human where in most cases the skin regions consist of one or two 16 by 16 macroblocks. Usually, the skin information from the DCT values of color components can not be used for human detection since the resolution requirement is not the adjacency information between nodes with overlapping boundaries or areas.

- **Relational Graph Matching:** The last part of the algorithm is based on a graph matching approach. Modeling human with relational graphs is widely used in the literature. However, most of them rely on satisfactory segmentation results. The meaningful combinations of classes is used to overcome this problem. In graph representation of human, each level of a branch represents a class for a body part or combination. Each body part and meaningful combinations represent a class. The combination of binary and unary features is represented by a feature vector. For the purpose of determining the class of these feature vectors a piecewise quadratic Bayesian classifier is used. In our case, it is a multiclass and multifeature problem. For the reference model supervised learning is implemented using several test images. The features for each body part are assumed to be Gaussian distributed.

4 Results

To evaluate the system performance for the activity recognition, we used several sequences with different activities. Table 1 displays the resulting normalized distances (Eq. 2) between the activity sets and test sequences. The preliminary results show that MPEG motion vectors corresponding to three human body subregions can be used for detection and recognition of human activity. Each test sequence gives the minimum normalized distance with its corresponding training set. The last sequence is a MPEG car movie. Note that the distances are very high for each activity class. Another restriction for car sequences is that the human body ratio is not suitable for the car mainbody. The performance of the algorithm depends on the temporal duration of the observed activity. The results displayed in the table are given for sequences with two or more activity periods. Different time instants during one period are detected in the second step. Some results are displayed in Figure 3.

Some of the human detection and posture recognition results are also given in this section. Since human body parts are smooth objects the smoothing factor is chosen small (= 1.25). Curvature threshold is chosen the same for all the test images (= 0.55). In Figure 6, the curvature segmentation result for selected body parts is shown.

	Walking	Running	Kicking
walk1	0.001	0.0587	0.1543
walk2	0.0103	0.0929	0.0615
walk3	0.007	0.02	0.0784
walk4	0.0084	0.1218	0.1627
walk5	0.046	0.1506	0.1651
walk6	0.019	0.1298	0.208
run1	0.26677	0.0954	0.1688
run2	0.2525	0.0143	0.2519
run3	0.7665	0.027	0.1703
kick1	0.298	0.1253	0.0576
kick2	0.1901	0.109	0.0868
car	0.5362	0.4282	0.6922

Table 1. The normalized Euclidean distances between the activity sets and test sequences.

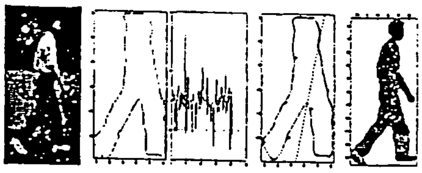


Figure 6. First: Frame from MPEG7 test sequence. Second: Leg segment. Third: Curvature of the segment ($th_k = 0.55$). Fourth: Curvature segmentation. Fifth: Segmentation result.

The performance of the graph matching algorithm is given for 42 test images for front and side views which are chosen from different sources. In the model file, the adjacency information between parts is given as; head-torso, upper arm-torso, leg-foot, lower arm-hand, etc. For example, there is no adjacency restriction between hand and leg or hand and belly, since hand can be at any position near them. In the model file these combinations are also chosen: arm=upper arm+lower arm, legs=leg1+leg2, lowbody=legs+belly, upbody=torso+belly, armtorso=arm+torso. Results for segmentation and modeling with superellipses are displayed in Figure 7.

After graph matching, the body parts in Figure 7 (first row); face, torso, belly, arm 1, arm 2, leg 1, and leg 2, are correctly classified. Face, torso, and legs (together) are the classified body parts for the second row where the person wears a suit which covers multiple body parts. In the third row, a side-view image is displayed. The correctly classified parts are, face, arm 1, torso, leg 1, leg 2, foot 1, and foot 2. The graph matching algorithm performance is obtained by computing the correct, false, and miss detection of the body parts in the test images. The preliminary results show that % 70.27 of the body parts are correctly and % 18.92 are falsely classified. The remaining % 10.8 is the miss detection. The majority of the falsely classified body parts are hand and foot regions that are generally combined with leg and arm regions. In order to determine the posture of the persons in the still images and video sequences, we use the binary features of the corresponding matched node pairs after the classification. For example, the angle α between the image node matched to torso and image node matched to arm informs how much arms are open. Table 2 displays the results

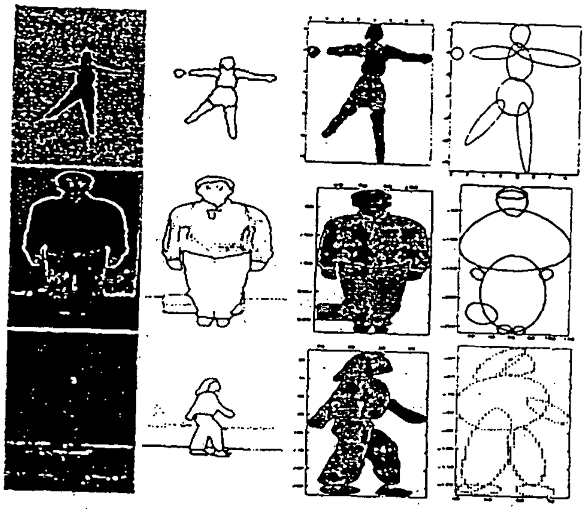


Figure 7. First column: Original images. Second column: Segmentation results. Third column: Part separation and curvature segmentation results. Fourth column: Fitted superellipses.

part 1	part2	α
torso	arm 1	79.10
torso	arm 2	75.32
torso	leg 1	39.31
torso	leg 2	2.92

Table 2. Relative orientation (α) values for Figure 7 (first row).

for Figure 7 (first row) where both arms are open with an angle of 75-80 degrees, one leg is open with an angle of 40 degrees while other leg is approximately on the same axis as torso. Note that, posture recognition is a direct result of correct classification of the body parts. The segmented body parts can be also used for a more detailed tracking and activity recognition in the uncompressed domain.

5 Conclusions

In this paper, we proposed a hierarchical method for human activity and posture recognition in MPEG sequences for different resolution levels from low complexity to low false rates. The preliminary results indicate that a significant information can be obtained from the compressed domain. Experimental results of principal component analysis show that macroblock motion vectors can be used for activity recognition if the observed activity consists of two or more periods. Detection of human skin regions in the compressed or uncompressed domains increases the performance of the proposed graph matching algorithm.

References

[1] I.B. Ozer, W. Wolf, A.N. Akansu, "Relational Graph Matching for Human Detection and Posture Recognition", SPIE Symposium on Voice, Video, and Data Communications, Nov. 2000, Proceedings of SPIE, Vol. 4210.

[2] S.-F. Chang, J. R. Smith, M. Beigi, and A. B. Benitez, "Visual Information Retrieval from Large Distributed On-line Repositories", Communications of the ACM, Vol. 40, No. 12, 1997, pp. 63-71.

[3] M.M. Yeung, B.L. Yeo, W. Wolf and B. Liu , "Video Browsing using Clustering and Scene Transitions on Compressed Sequences", SPIE Vol. 2417 Multimedia Computing and Networking 1995 , pp. 399-413.

[4] B.L. Yeo and B. Liu , "On the extraction of DC sequence from MPEG Compressed Video", IEEE ICIP, Oct. 1995.

[5] A. M. Dawood and M. Ghanbari, "Scene Content Classification From Mpeg Coded Bit Streams", IEEE Workshop on Multimedia Signal Processing, 1999, pp 253-258.

[6] Kadir A. Peker, A. Aydin Alatan, Ali N. Akansu, "Low-level Motion Activity Features for Semantic Characterization of Video", Proceedings of IEEE International Conference on Multimedia and Expo, 2000.

[7] D. Schonfeld and D. Lelescu, "VORTEX: Video retrieval and tracking from compressed multimedia databases - template matching from MPEG2 video compressed standard", SPIE Conference on Multimedia and Archiving Systems III, Nov. 1998.

[8] H. Wang and Shih-Fu Chang, "A Highly Efficient System for Automatic Face Region Detection in MPEG Video Sequences", IEEE Trans. on Circuits and Systems for Video Technology, special issue on Multimedia Systems and Technologies, Vol. 7, No. 4, Aug. 1997, pp. 615-628.

[9] H. Wang, H. S. Stone, and S.-F. Chang, "FaceTrack: Tracking and Summarizing Faces from Compressed Video", SPIE Multimedia Storage and Archiving Systems IV, 19-22 Sept, Boston".

[10] Y. Yacoob and M. J. Black, "Parameterized Modeling and Recognition of Activities", ICCV, 1998, pp120-127.

[11] J.K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review," Computer Vision and Image Understanding, Vol.73, No.3, pp. 428-440, March 1999.

[12] D.M. Gavrila, "The Visual Analysis of Human Movement: A Survey", Computer Vision and Image Understanding, Vol.73, No.1, pp. 82-98, January 1999.

[13] M. Walter, S. Gong, A. Psarrou, "Learning Stochastic Temporal Models of Human Behaviour", Proc. IEEE International Workshop on Modelling People, Corfu,1999.

[14] K. Rangarajan, W. Allen, M. Shah, "Matching Motion Trajectories Using Scale-Space", Pattern Recognition, Vol 26, No 4, pp 595-610.

[15] R. Cutler and L. Davis, "Real-Time Periodic Motion Detection, Analysis and Applications", CVPR, 1999, pp. 326-332.

[16] C. Curio, J, Edelbrunner, T. Kalinke, C. Tzomakas, W. von Seelen, "Walking Pedestrian Recognition", ITSC, 1999, pp. 292-297.

[17] M. Kirby and L. Sirovich, "Application of the Karhumen-Loeve Procedure for the Characterization of Human Faces", IEEE PAMI, Vol 12(1), 1990, pp.103-108.

[18] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces", CVPR 1991, pp. 586 -591.

Relational Graph Matching for Human Detection and Posture Recognition

I. Burak Ozer[†], Wayne Wolf[‡], Ali N. Akansu[†]
[†]Department of Electrical and Computer Engineering, New Jersey Institute of Technology,
New Jersey Center for Multimedia Research, Newark, NJ 07102, USA
[‡]Department of Electrical Engineering, Princeton University, Princeton, NJ 08540, USA

ABSTRACT

This paper describes a relational graph matching with model-based segmentation for human detection. The matching result is used for the decision of human presence in the image as well as for posture recognition. We extend our previous work for rigid object detection in still images and video frames by modeling parts with superellipses and by using multi-dimensional Bayes classification in order to determine the non-rigid body parts under the assumption that the unary and binary (relational) features belonging to the corresponding parts are Gaussian distributed. The major contribution of the proposed method is to create automatically semantic segments from the combination of low level edge or region based segments using model-based segmentation. The generality of the reference model part attributes allows detection of human with different postures while the conditional rule generation decreases the rate of false alarms.

Keywords: Human detection, posture and activity recognition, graph matching, model-based segmentation.

1. INTRODUCTION

Great effort has been devoted to human recognition related topics such as face recognition in still images and motion analysis of human body parts. Most of the previous work depend highly on the segmentation results and mostly motion is used as a cue for segmentation.[1] There has been very few work that are on the human recognition in still images. Although in[2,3] the authors use a compact representation of the training sets that are suitable for cluttered scenes there is no direct correspondence between the low level features and body parts. Such a semantic representation is needed for high level applications and for occlusion problems. In another survey,[4] the segmentation problem is again pointed out especially for the detection of multiple and occluded humans in the scene. The problem of connecting low level features to high level semantics via a graph based description scheme in similarity retrieval was addressed in our previous work[5] for rigid body (i.e. car). Compared to other methods that use relational model for human body, the major contribution of the proposed method is to create automatically semantic segments from the combination of low level edge or region based segments using model-based segmentation. High level semantics do not only help in the detection stage but also in posture recognition since relational graph matching results can describe the movement of the person without any further computation. Another advantage is that it is easily extended to human tracking in video frames. In this paper, we propose an improved method (Figure 1) with parametric modeling of the segmented body parts by using superellipses and Bayes classification framework in the graph matching process. Superquadrics are widely used for modeling three dimensional objects in computer vision literature.[6,7] In[8] and,[9] the authors use single test objects with a uniform background, and model object subparts with 2D superquadrics. In our case, we try to detect multiple human in different images. Even when human body is not occluded by another object, due to the possible positions of non-rigid parts, a body part can be occluded in different ways. Parametric modeling of image segments helps to overcome this problem and reduce the effect of the deformations due to the clothing. Graph matching allows to increase the generality of the reference model part attributes for the detection of human with different postures while the conditional rule generation between graph levels decreases the rate of false detection.

Some preliminary results for a prototype system,[10] that uses a hierarchical method for human detection and activity recognition in MPEG sequences, are also presented. The last step of this system is based on the proposed graph matching algorithm.

The paper is organized as follows: in the second section the overall algorithm is presented. Curvature segmentation and modeling with superellipses are given in section three and four respectively. Section five describes graph matching procedure. In section six, we give a brief description of the proposed MPEG system. Results are displayed in section seven.

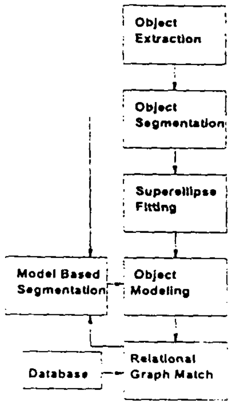


Figure 1. Overall algorithm.

2. ALGORITHM

The algorithm is illustrated in Figure 1. The input is a still image or a video sequence, and the output is extraction of the human body region with the labeled parts.

- **Object Extraction:** This part corresponds to video applications. Separation of a moving object in a video is covered here. In the uncompressed video sequences, we track the feature points of an object using the Kanade-Lucas-Tomasi tracking method[11] and group them according to their moving directions and distances. Only the feature points with a velocity greater than a given threshold are considered. Next is the determination of a rectangular region of interest by calculating the center of gravity and the eccentricity of these groups. For the MPEG compressed video sequences, motion vectors of macroblocks are used for the extraction of the moving object.

- **Object Segmentation:** An object usually contains several sub-objects (head, torso, hands, etc.) that can be obtained by segmenting the object of interest (OOI) hierarchically into its smaller unique parts. We use the color image segmentation technique proposed in[12] combined with an edge detector algorithm. Skin color model is formed via Farnsworth nonlinear transformation. The segmented image can contain regions corresponding to the background. However, these regions will not match the regions of the template object. The segmented region boundaries can still be in complex forms. The boundaries are first smoothed by a Gaussian smoothing operator. Concave and convex segments (landmarks) that are used for curvature segmentation are then determined on the resulting contour. Curvature segmentation is explained more detailed in the following section.

- **Superellipse Fitting:** Each segmented region is modeled with a superellipse. Even when human body is not occluded by another object, due to the possible positions of non-rigid parts a body part can be occluded in different ways. For example, hand can occlude some part of torso or legs. The contour approximation, used in our previous work,[5] is not efficient in this case and the combination of occluded part with hand is not meaningful. In this case, 2D approximation of parts by fitting superellipses with shape preserving deformations provides more satisfactory results. It also helps to discard the deformations due to the clothing. Global approximation methods give more satisfactory results for human detection purposes. Hence, instead of region pixels, parametric surface approximations are used to compute shape descriptors. A detailed superellipse description and fitting procedure are given in section 4.

- **Object Modeling and Similarity Measure:** Geometric descriptors for simple object segments such as area, compactness (circularity), weak perspective invariants,[13] and spatial relationships are computed. These descriptors are classified into two groups: unary and binary features.

Unary features:

a) compactness; b) eccentricity; c) color (hair and skin).

To represent the skin and hair color, perceptually uniform color system (UCS), proposed by Farnsworth[14,15] is used. Like other attributes, color attribute (c_j) of an image segment will be separated by a distance from the model color (c_i) with tritimulus values (t_1, t_2, t_3). This color difference measure must reflect noticeable color differences in order to capture skin and hair color models and still be feasible to work in Euclidean space. Farnsworth nonlinear transformation produces uniform noticeable color differences that can be used in this approach. First RGB color information is converted to XYZ color system and the resulting chromaticity components are transformed using Farnsworth nonlinear transformation to the new chromaticity (u, v) values. The noticeable color differences in the XY chromaticity diagram can be fitted by ellipses, but these color differences become much more circular and tend to be uniform in the UV diagram.[15] These (u, v) values and the luminance are used to determine skin and hair locations in the image with adjacency and shape attributes (Figure 3). Our method relies mainly on the skin color model since the hair color model is not that reliable.

Binary features:

a) Ratio of areas; b) relative position and orientation; c) the adjacency information between nodes with overlapping boundaries or areas. The relative position and orientation (Figure 2) are computed using the weak perspective approximation[13]:

$$\begin{aligned} u &= \frac{(\vec{p_3} - \vec{p_1}).(\vec{p_2} - \vec{p_1})}{|\vec{p_2} - \vec{p_1}|^2} \\ v &= \frac{(\vec{p_3} - \vec{p_1}).(\vec{p_2} - \vec{p_1})^{\perp}}{|\vec{p_2} - \vec{p_1}|^2} \\ \cos(\alpha) &= \frac{(\vec{p_2} - \vec{p_1}).(\vec{p_4} - \vec{p_3})}{|\vec{p_2} - \vec{p_1}||\vec{p_4} - \vec{p_3}|} \end{aligned}$$

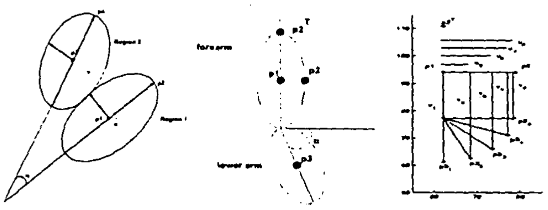


Figure 2. Left: Relative position(RP) and orientation(OR) of two regions. Middle: Arm model. Right: RP and OR changes of the forearm and lower arm with respect to each other.

• **Relational Graph Matching with Model-Based Segmentation:** The last part of the algorithm is based on a graph matching approach. Modeling human with relational graphs is widely used in the literature. However, most of them rely on satisfactory segmentation results. The meaningful combinations of classes is used to overcome this problem. A "meaningful" combination is the combination of adjacent segments on the same principle axis. For example, upper arm of a person with a shirt can be segmented into two parts, however it should be the combination of clothed and naked regions. The opposite of this example can also occur e.g., color and curvature segmentation can fail to segment arms from torso (Figure 6). Hence, in graph representation of human, each level of a branch represents a class for a body part or combination.

3. CURVATURE SEGMENTATION

The contour points of each segment is smoothed by a Gaussian smoothing operator to reduce the effect of image noise and clothing. Then, the concave points with high curvature (greater than a threshold th_k) and arc lengths (greater than a threshold th_s relative to the segment length) are marked:

$$u(K_s - th_k)\delta(\frac{dK_s}{dt} - 0)u(s_{K_s}(t) - th_s) = 1 \qquad (1)$$

where t is the arc length parameter, $s_{K_s}(t)$ is the length of concave arc, u is the step function and δ is the delta function. A normal line is computed from this landmark until it reaches another point on the contour. Then, the segment is divided at these points and an interpolation is performed between these points to form closed segments.

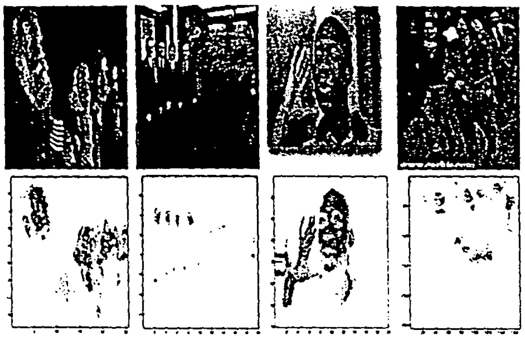


Figure 3. Skin color segmentation results for some test images.

As expected, experimental results show that the high curvature locations occur at the joints on the limbs. Since human body parts are smooth objects the smoothing factor is chosen very small (= 1.25). Curvature threshold is chosen the same for all the test images (= 0.55). In Figure 4, the curvature segmentation results for the selected body parts are displayed. Note that, since the arc length at the junction of the legs (belly) is small relative to the whole segment length, this part is not segmented. The graphs, given in Figure 4, show the curvature points. For the arm segment, there is one concavity point which is greater than the curvature threshold while for the leg segment, all the concave points are below this threshold. Figure 5 displays another example from a MPEG7 test sequence.

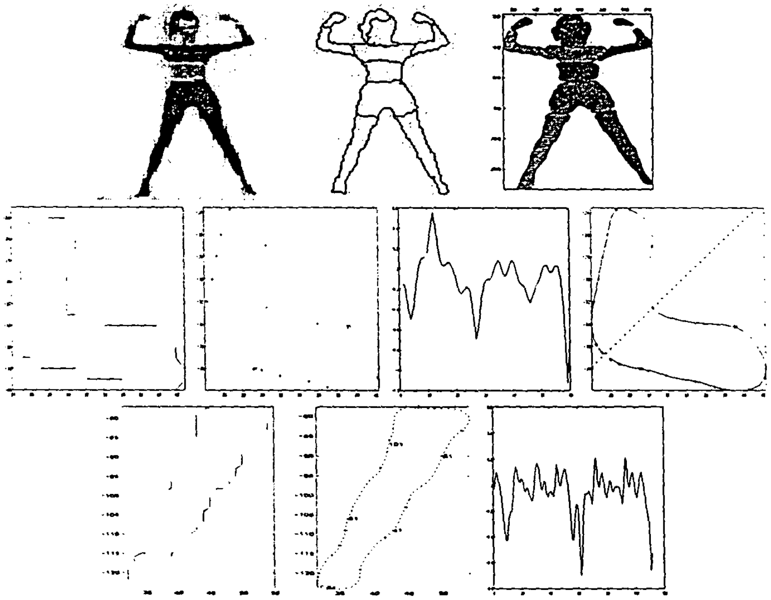


Figure 4. Top: First: Original image. Second: Segmentation result. Third: Curvature segmentation results. Middle: First: Arm segment. Second: Smoothed contours with landmarks (th_k = 0.55). Third: Curvature points. Four: Curvature segmentation. Bottom: First: Leg segment. Second: Smoothed contours with landmarks (th_k = 0.55). Third: Curvature points.

4. MODELING WITH SUPERELLIPSES

Each segmented region is modeled with a superellipse. A superellipse[6,7] can be described explicitly as:

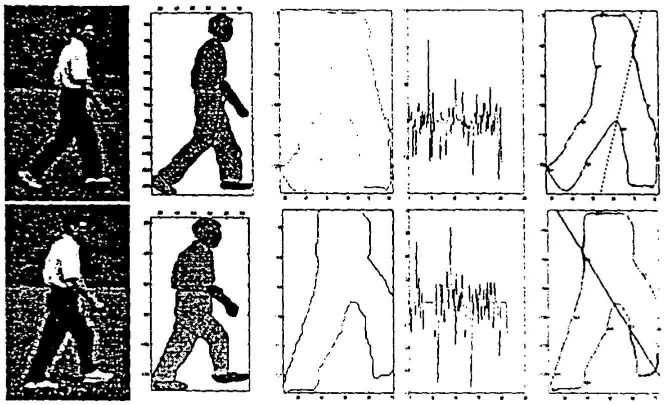


Figure 5. First column: KLT algorithm result for the MPEG7 test sequence. Second column: Segmentation results. Third column: Leg segment. Fourth column: Curvature of the segment($th_k = 0.55$). Fifth column: Curvature segmentation.

$$x = f_x(\eta) = a_x cos^\epsilon(\eta), \quad y = f_y(\eta) = a_y sin^\epsilon(\eta)$$

where $-\pi < \eta < \pi$, a_x and a_y are two semi-axis, and ϵ is the roundness parameter. The inside-outside function of a two dimensional superquadric can be given as:

$$(\frac{x}{a_x})^{2/\epsilon} + (\frac{y}{a_y})^{2/\epsilon} = f(x, y, \mathbf{a}) \tag{2}$$

where **a** is the parameter set.

There can be various deformations that can be implemented on the superellipses. Tapering and bending are sufficient deformations to represent human body. However, when for example legs are wide open we have to segment the legs since no shape preserving deformation can represent them. Tapering along the y-axis is computed as:

$$X = (\frac{K}{a_y} + 1)x, \quad Y = y$$

where K is a constant. Circular bending is computed as:

$$\begin{aligned} X &= x + sign(b)(\sqrt{y^2 + (a_y/b - x)^2} - (a_y/b - x)) && (3) \\ Y &= sin(atan(y/(a_y/b - x))(a_y/b - x) && (4) \end{aligned}$$

In these equations, b is the bending parameter, (X, Y) are the transformed (x, y) values where the transformation is represented as $(D \circ R \circ T)(x, y) \rightarrow (X, Y)$ with D = Deformation, R = Rotation (θ), T = Translation (p_x, p_y). In order to find superellipse parameter set $\mathbf{a} = [a_x, a_y, \epsilon, K, b, \theta, p_x, p_y]$ that fits best to the segment data (X, Y) we use Levenberg-Marquardt method.[16]

First, the initial parameter set is used to find nondeformed world centered superellipse $(\bar{x}, \bar{y})$

$$(D \circ R \circ T)^{-1}(X, Y) \rightarrow (\bar{x}, \bar{y}) \tag{5}$$

(6)

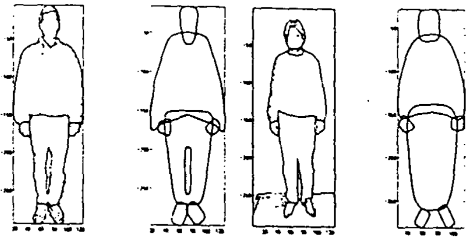


Figure 6. Approximations for two bodies.

The model to be fitted, the inside-outside function $f(\overline{x},\overline{y},\mathrm{a})$ forms the merit function χ in order to determine best fit parameters by its minimization.

$$(\frac{\overline{x}}{a_x})^{2/\epsilon} + (\frac{\overline{y}}{a_y})^{2/\epsilon} = 1 \qquad (7)$$

$$\chi^2(\mathrm{a}) = \sum_{i=1}^{N}(1 - f(\overline{x},\overline{y},\mathrm{a}))^2 \qquad (8)$$

The initial parameter set is taken as following:

$$\epsilon = 1, \quad K = 0, \quad b = 0, \quad px = 1/N\sum_{i=1}^{N} X_i, \quad py = 1/N\sum_{i=1}^{N} Y_i$$

The orientation of the object is estimated by its second order moments. Some superellipse examples are displayed in Figure 6.

5. GRAPH MATCHING

Detection of objects is achieved by matching the relational graphs of objects (S regions) with the reference model. The input image graph O_n with N nodes ($N \geq S$) and a reference graph (O_r with N_r nodes) are matched. The aspect graph of the reference object is formed according to the segmentation results of the training images. Two reference models namely front and side view models are used in the experiments. Our assumption is that human face (at least a part of it) must be seen since skin color is a dominant attribute for head (hair color model is also used but it is not that reliable)(Figure 7). Face detection allows to start initial branches efficiently and reduces the complexity. B_h represents the group of branches for the corresponding head area. Note that false face detection will result in a branch with single or very few matched nodes and will be eliminated. Relational graph matching would allow human detection without face part however it would increase the computational complexity significantly.

Each body part and meaningful combinations represent a class (ω). The combination of binary and unary features is represented by a feature vector (X). Note that feature vector elements change according to body part and the nodes of the branch under consideration. For example, for the first node of the branch, feature vector consists of unary attributes. The feature vector of the following nodes includes also binary features dependent on the previous matched nodes in the branch. In order to determine the class of these feature vectors a piecewise quadratic Bayesian classifier is used. In our case, it is a multiclass and multifeature problem. For the reference model supervised learning is implemented using several test images. The features for each body part are assumed to be Gaussian distributed. From Bayes theorem:

$$k = arg\max_j P(\omega_j|X) = \max_j \frac{p(X|\omega_j)P(\omega_j)}{p(X)} \rightarrow X \in \omega_k$$

where $P(\omega_j)$ is a priori probability, $P(\omega_j|X)$ is a posteriori probability and ω represents a class. We can write the discriminant function as[17]

$$g_j(X) = log(p(X|\omega_j)) + log(P(\omega_j))$$

For multifeature problems with arbitrary covariance the decision surfaces are hyperquadrics and the resulting discriminant functions are

$$g_j(X) = X^T W_j X + \omega_j^T X + \omega_{j0}$$

where

$$W_j = -1/2\Sigma_j^{-1}, \quad \omega_j = \Sigma_j^{-1} M_j$$
$$\omega_{j0} = -1/2 M_j^T \Sigma_j^{-1} M_j - 1/2 log|\Sigma_j| + log P_{\omega_j}$$

where M_j represents the class mean and Σ_j is the covariance matrix of each class.

During supervised learning, for each reference model node that represents a class $p(X|\omega_j)$ is computed. $P(\omega_j)$ is computed with the assumption that each class is equal probable and parts such as arms represent two classes in the model file. Note that our problem differs from the classical Bayes classification method in the sense that we do not try to find the class of a given feature vector by minimizing the risk factor but we try to find the existence of a member for a given class. We want to find if the OOI exists in the image by matching the image segments to possible classes of OOI. Due to the generality of the problem "detecting human" and high variance of the within-class scatter matrices of unary feature vectors for different body parts, the relational features must be used. Furthermore, conditional rule generation (r) eliminates the image segments that do not hold human body rules such as "face must be adjacent to torso", "if two arms are already matched in the branch there can not be another arm classification for that branch", and "angle between torso and face principal axis (α) can not exceed a certain threshold". Hence our problem is to find the existence of a member among image segments of a model class by maximizing the probability of feature vector for the given class in the corresponding branch.

The overall algorithm for the relational graph matching is given below.

```
for every model node $j \in O_r$ do
    for every branch $b$ do
     $(i_1, i_2)$ =match($j, b$)
     copy branch $b$ and add node pair $(j,i_1)$ in the new branch and update $G^b$ by adding $g_j^b(X_{i_1})$
     copy branch $b$ and add node pair $(j,i_2)$ in the new branch and update $G^b$ by adding $g_j^b(X_{i_2})$
    end for
end for
choose $arg\max_{b \in B_k} G^b$

match($j, b$)
for every image node $i \in O_n$ do
    for every matched node pair $(b_j, b_i)$ in the branch do
        if $\exists r(b_j, j)$ then
```

```
            if $r(b_i, i)$ holds then
                compute $g_j^b(X_{i,b_i})$
            else
                $g_j^b(X_{i,b_i}) = 0$
            end if
        else
                compute $g_j^b(X_i)$
        end if
    end for
end for
Return image nodes $i_1, i_2$ with two highest $g_j^b(x_i)$ values > threshold
```

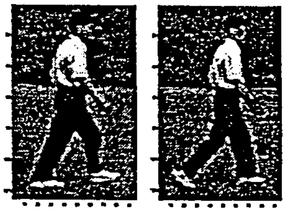


Figure 7. Superellipse fitting to the detected skin parts.

6. HUMAN ACTIVITY DETECTION IN MPEG SEQUENCES

In this section, a possible use of the proposed graph matching algorithm in a prototype human detection system,[10] is given. The system can be divided into three major parts. The first part is activity recognition in the compressed domain based on principal component analysis (PCA).[18] The data sets for human detection applications in the literature for the compressed domain include anchorperson scenes, news stories and interviews, where the faces and the upper-bodies occupy large areas in the image. However, at lower resolution, available motion vectors can be used to detect human activity by comparing it with known human activity patterns. In this method, we first detect the moving regions and then group the motion vectors automatically by using the ratio of the human body parts. Hence the measurements do not correspond to the actual human body parts but to macroblock groups corresponding to human region. For the classification of moving regions, the neighboring blocks with a velocity greater than a predefined threshold are classified as one moving object. To evaluate the system performance for the activity recognition, we used several sequences with different activities. Table 1 displays the resulting normalized distances between the activity sets and test sequences.

In the second part, DC DCT differences between frames in the compressed domain are matched to activity templates (side-view), obtained from a training set, to distinguish activity periods. The DC coefficients are also used in the graph matching algorithm for human body recognition in the compressed domain, but this method is suitable for images with face regions greater than 3 by 3 macroblocks. Since graph matching performance depends highly on face detection, this is a crucial restriction. In most cases, the resolution of the face area does not satisfy this criteria, which leads us to implement the graph matching algorithm in the uncompressed domain for the finest analysis of the human body and posture.

	Walking	Running	Kicking
walk1	0.001	0.0587	0.1543
walk2	0.0103	0.0929	0.0615
walk3	0.007	0.02	0.0784
walk4	0.0084	0.1218	0.1627
walk5	0.046	0.1506	0.1651
walk6	0.019	0.1298	0.208
run1	0.26677	0.0954	0.1688
run2	0.2525	0.0143	0.2519
run3	0.7665	0.027	0.1703
kick1	0.298	0.1253	0.0576
kick2	0.1901	0.109	0.0868
car	0.5362	0.4282	0.6922

Table 1. The normalized Euclidean distance between the activity sets and test sequences.

7. RESULTS

In this section, the performance of the proposed graph matching algorithm (Figure 1) is given for 42 test images for front and side views which are chosen from different sources. Since bending deformation increases the computational complexity and curvature segmentation is used, the computations are done using only the tapering deformation. An example model file is shown in Figure 9. In the model file, the adjacency information between parts is given as; head-torso, upper arm-torso, leg-foot, lower arm-hand, etc. For example, there is no adjacency restriction between hand and leg or hand and belly, since hand can be at any position near them. In the model file these combinations are also chosen: arm=upper arm+lower arm, legs=leg1+leg2, lowbody=legs+belly, upbody=torso+belly, armtorso=arm+torso. Another important issue in the model file generation is that the features, such as eccentricity, can show large deviations from person to person (thin-fat, big-small, etc.) for each body part. Furthermore, eccentricity of the limbs are close to each other. Hence, within-class scatter matrix can be large while between-class scatter matrix can be small which is the worst case for a classification. Under the assumption that feature vectors have Gaussian distribution, we determine their mean and variance during supervised learning. Figure 8 displays the circularity and eccentricity distributions for face.

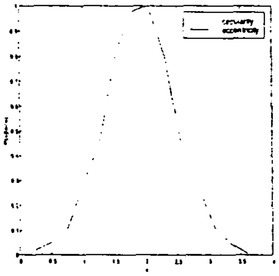


Figure 8. Distributions of two face features.

Results for segmentation and modeling with superellipses are displayed in Figure 10 for different test images. Classification results for three images in Figure 11 are given in Table 2. Note that, in Figure 11 d), an image with multi-persons is tested. Since the algorithm first determines the face regions, different branches for possible face regions are initialized. In the same image, the lower arms of the persons are folded on their upper arms where graph matching algorithm classifies them as upper arms. The overall algorithm performance is obtained by computing the correct, false, and miss detection of the body parts in the test images. The preliminary results show that % 70.27 of the body parts are correctly and % 18.92 are falsely classified. The remaining % 10.8 is the miss detection. The majority of the falsely classified body parts are hand and foot regions that can not be properly segmented from arms and legs.

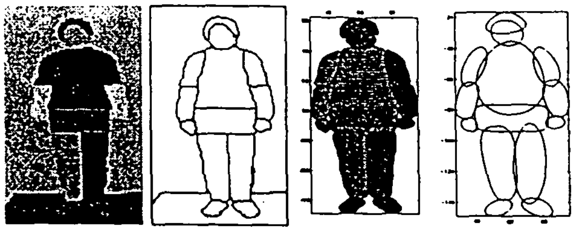


Figure 9. First: The skin areas are determined in the model color image. Second: Segmentation result. Third: Curvature segmentation results. Four: Fitted superellipses to the body parts.

In order to determine the posture of the persons in the still images and video sequences, we use the binary features of the corresponding matched node pairs after the classification. For example, the angle α between the image node matched to torso and image node matched to arm informs how much arms are open. Table 3 displays an example where both arms are open with an angle of 75-80 degrees, one leg is open with an angle of 40 degrees while other leg is approximately on the same axis as torso. Tables 4 and 5 display the angles between torso-arms and torso-legs for the multi-person image. Since the angles are very small, it can be easily determined that both of the persons have closed arms and closed legs where their arms and legs are approximately on the same axis of torso. Note that, posture recognition is a direct result of correct classification of the body parts.

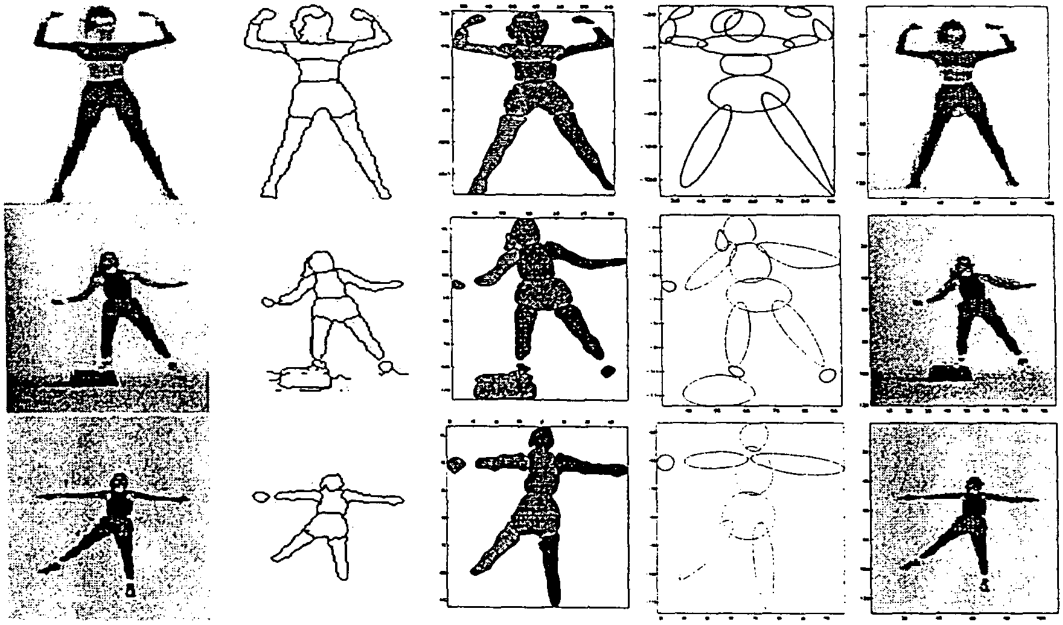


Figure 10. Column 1: Original images. Column 2: Segmentation results. Column 3: Part separation and curvature segmentation results. Column 4: Fitted superellipses. Column 5: Indexed superellipses on the body where superellipses with the skin areas are also determined.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we extended our previous work for human detection in still images and video frames by using superellipses and Bayes classification in the relational graph matching algorithm. This method enables the detection of people in the scene as well as posture recognition by using model-based segmentation. The future work will focus on the occluded images and video frames with multiple persons. Our current work covers the improvement of the proposed human activity detection system for MPEG sequences.

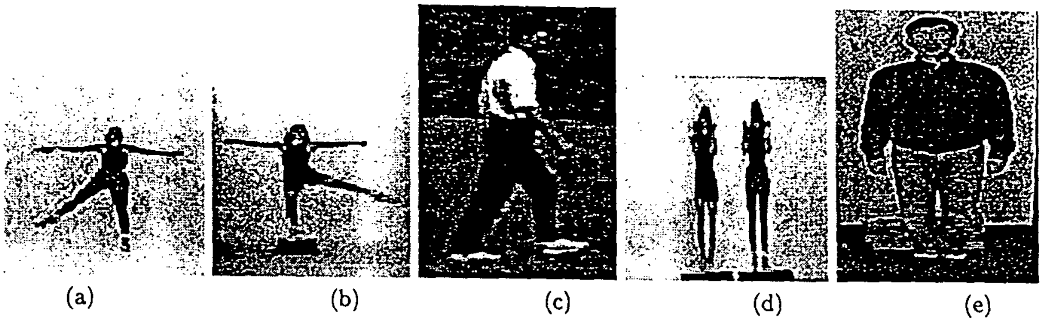


Figure 11. Some test images. The detection performance for image a), d) and e) are given in Table 2.

model - image(a)	model - image(d)	model - image(e)
face - face	face - face(Right body)	face - face
torso - torso	torso - torso(")	torso - torso
belly - belly	belly - belly(")	legs - legs
arm1 - arm1	uparm1 - lowarm1(")	
arm2 - arm2	uparm2 - lowarm2(")	
leg1 - leg1	leg1 - leg1(")	
leg2 - leg2	leg2 - leg2(")	
	face - face(Left body)	
	torso - torso(")	
	belly - belly(")	
	uparm1 - lowarm1(")	
	uparm2 - lowarm2(")	

Table 2. Classification results for three test images.

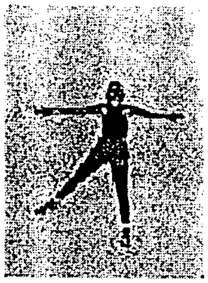


Figure 12.

part 1	part2	α
torso	arm 1	79.10
torso	arm 2	75.32
torso	leg 1	39.31
torso	leg 2	2.92

Table 3. α values ($\alpha = \Delta\theta$)

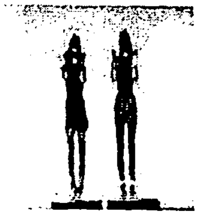


Figure 13.

part 1	part2	α
torso	arm 1	7.94
torso	arm 2	9.10
torso	leg 1	5.11
torso	leg 2	6.12

Table 4. α values for the person on the left.

part 1	part2	α
torso	arm 1	1.98
torso	arm 2	2.92
torso	leg 1	0.81
torso	leg 2	0.82

Table 5. α values for the person on the right.

REFERENCES

1. J.K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review," *Computer Vision and Image Understanding*, Vol.73, No.3, pp. 428-440, March 1999.
2. U. Franke and D. Gavrila, "Autonomous Driving Goes Downtown," *IEEE Intelligent Systems*, Vol.13, No.6, pp. 40-48, November 1998.
3. C. Papageorgiou, M. Oren and T. Poggio, "Pedestrian Detection Using Wavelet Templates," *Proc. of CVPR*, Puerto Rico, June 1997.
4. D.M. Gavrila, "The Visual Analysis of Human Movement: A Survey," *Computer Vision and Image Understanding*, Vol.73, No.1, pp. 82-98, January 1999.
5. I. B. Ozer, W. Wolf, and A. N. Akansu, "A Graph Based Object Description for Information Retrieval in Digital image and Video Libraries", CBAIVL, pp.79-83, 1999.
6. A.H. Barr, "Superquadrics and Angle Preserving Deformations," *IEEE Computer Graphics Applications*, Vol.1, pp. 11-23, 1981.
7. Solina, F.; Bajcsy, R., "Recovery of parametric models from range images: the case for superquadrics with global deformations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.12, No.2, pp. 131-147, Feb. 1990.
8. M. Bennamoun, R. Boashash, "A Vision System for Automatic Object Recognition," *Proc. of IEEE International Conference on Systems, Man, and Cybernetics, 1994. Humans, Information and Technology.*, Oct. 1994.
9. M. Bennamoun, B. Boashash, "A Structural-Description-Based Vision System for Automatic Object Recognition", IEEE Transactions on Systems, Man, and Cybernetics-Part B, Cybernetics, Vol 27, No 6, December 1997.
10. I. B. Ozer, W. Wolf, and A. N. Akansu, "Human Activity Detection in MPEG Sequences", Submitted, July 2000.
11. J. Shi and C. Tomasi, "Good Features to Track", *CVPR*, 1994.
12. K. Harris, S.N. Efstratiadis, N. Maglaveras, and A.K. Katsaggelos, "Hybrid Image Segmentation Using Water Sheds and Fast Region Merging", *IEEE Trans. on Image Processing*, vol. 7, pp.1684-1699, 1998.
13. J. B. Burns, R. S. Weiss and E. M. Riseman, "View Variation of Point-Set and Line-Segment Features", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, No 1, pp. 51-68, 1993.
14. H. Wu, Q. Chen, and Y. Yachida, "Face Detection From Color Images Using a Fuzzy Pattern Matching Method", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, No 6, pp. 557-562, 1993.
15. W.K. Pratt, "Digital Image Processing", J. Wiley and Sons, Second Edition, 1991.
16. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, "Numerical Recipes in C ", Cambridge University Press, Second Edition, 1995.
17. R.O. Duda, and P.E. Hart, "Pattern Classification and Scene Analysis ", John Wiley and Sons, 1973.
18. Y. Yacoob and M. J. Black, "Parameterized Modeling and Recognition of Activities", ICCV, 1998, pp120-127.

A Graph Based Object Description for Information Retrieval in Digital Image and Video Libraries

Burak Ozer[†] Wayne Wolf[‡] Ali N. Akansu[†]

†Department of Electrical and Computer Engineering
New Jersey Institute of Technology
New Jersey Center for Multimedia Research
Newark, NJ 07102, USA
ibo8175@oak.njit.edu, ali@megahertz.njit.edu

‡Department of Electrical Engineering
Princeton University
New Jersey Center for Multimedia Research
Princeton, NJ 08540, USA
wolf@ee.princeton.edu

Abstract

The search algorithms for the objects of interest related to shape similarity in a video or image library were implemented by various research groups. This work focuses on the search of a sample object (car) in video sequences and images related to the shape similarity. We also investigate a new description for cars, using relational graphs. The goal of this study is to investigate the shape matching method based on relational graph of objects with respect to its accuracy, efficiency and scalability. The aim is to annotate the images where the object of interest (OOI) is present. Then the query by text can be performed to extract images of OOI from a preprocessed database. The graph based description of the object with its meaningful parts provides an efficient way to obtain high level semantics from low level features. The hierarchical segmentation increases the accuracy of the detection of the object in the transformed and occluded images.

1 Introduction

Recently, the content based image and video retrieval has been of a great interest due to the MPEG-7 standard related activities. Since some visual properties of images, that are described by feature vectors, are difficult to describe with text, the similarity retrieval utilizing these vectors becomes important. Content based image indexing and retrieval has attracted great research attention at the governmental [6, 7] and industrial [8, 9] sites as well as at the universities [1, 2, 3, 4, 5, 10, 11] that use different techniques based on some features like shape, color, texture or a combination of them. The search algorithms for the objects of interest related to shape similarity in a video or image library were implemented by various research groups. Template matching [13], techniques using B-spline based modal matching [14], Fourier descriptors [15], and moment invariants [16] are the popular ones.

This work focuses on the search of a sample object (car) in video sequences and images related to the shape similarity. We also investigate a new description for cars, using relational graphs [17, 18]. The goal of this study is to investigate the shape matching method based on relational graph of objects with respect to its accuracy, efficiency and scalability. The aim is to annotate the images where the object of interest (OOI) is present. Then the query by text can be performed to extract images of OOI from a preprocessed database. The performance of the retrieval systems is not satisfactory due to the gap between high level concepts and low level features. The graph based description of the object with its meaningful parts provides an efficient way to obtain high level semantics from low level features. The hierarchical segmentation increases the accuracy of the detection of the object in the transformed and occluded images. The hierarchy level of the descriptions is scalable enabling solutions to different queries as "find a car" and "find a sports car". We provide an overview of the proposed method in section 2. The results are given in the third section. Last section includes the conclusion and future work.

2 Proposed Method

The proposed method is outlined in Figure 8. First step is the separation of a moving object in a video (or foreground object in an image [12]) (B2 in Fig.8). In a video sequence, we track the feature points of an object using the Kanade-Lucas-Tomasi tracking method [20] and group them according to their moving directions and distances. The next step is the determination of a rectangular region of interest by calculating the center of gravity and the eccentricity of these groups. The result of this step is a rectangular re-

Step 2: Increase j by 1. For every $i = 1, ..., N$ compute the matching cost between j^{th} and i^{th} node:

Matching cost = Unary feature differences between nodes j and i + Binary feature differences

Binary feature differences are computed according to the previously matched nodes for every branch: For every matched node pair (n_j, n_i) the relative area, perimeter, position and connectivity are computed between nodes j and n_j and between nodes i and n_i. The total matching cost is the weighted sum of these distances.

Step 3: If the matching cost between nodes j and i for a branch is smaller than a threshold found in the training process, set (j, i) as the new matched node pair for this branch. Note that i must be different from the previous n_i's.

Step 4: If all the $j = 1, ..., R_v$ nodes are executed, choose the branch with the maximum number of matched nodes. If there are more than one resulting branches, choose one with the smallest total matching cost.

Step 5: If majority of the reference graph nodes (%80) are not matched, go to Step 1 and repeat Steps 1-4 for combined nodes of the view class.

Step 6: If majority of reference graph nodes (%80) are matched, decide the presence of OOI, otherwise go to Step 1 for another view class and repeat Steps 1-5 until a match is found or all view classes are executed.

3 Results

The segmentation of an object into its regions and the computation of the attributes of each region form the low-level description of the object. These low-level descriptions of the object can be used for similar object retrieval in a database by graph matching that allows to associate with high level semantic concepts. The algorithm is implemented on the still images with OOI at the foreground and center of the image, and on video sequences with moving OOI. The object detection is done off-line for text annotation of images that contain OOI.

The subgraphs for three view classes shown in Figure 2 are obtained using the training images. The side-view class is given as an example. The nodes 1, 2 and 3 are the main body parts. Since the color of these parts are in general the same, these parts are not segmented after the color segmentation. The curvature segmentation divide the main body at the concave points above the tires. However, when the segmentation fails, the main body in a given input image may correspond to one node in the graph of this image. Hence, the combination of these nodes are further used if the matching of these individual nodes fails. Node 4 is the side window(s). Node 5 and 6 are the tires. It is observed that the tires can be adjacent to the main body or there can be other nodes between them corresponding to the shadowing part. The relative position, orientation and the adjacency of nodes form the arcs (a,..., g) between nodes. Around fifty car images have been segmented and a training set is formed by using manually extracted car parts. Each part corresponds to a node in a view class. Mean and variance of the node attributes are computed to obtain an optimum reference model.

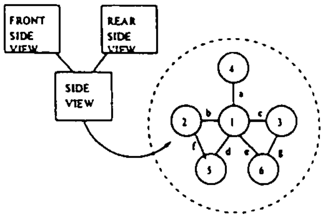


Figure 2. Reference graph model.

The detection of car images is implemented on MPEG 7 sequences (MPEG-7 Video Content Set (Category: Surveillance, Type: Shot, Item num.: V29, Description: Video sequences taken from a bridge over a speedway, Source: UCL)), on still images (A) with uniform background and still images (B) with nonuniform background and cars with many colors on the main body, causing poor object segmentation (Figure 4)[1]. An example showing the processing steps is displayed in Figure 3. The results summary is listed in Table 1. Note that, the segmentation of object in sequences is more successful since motion is incooperated in the segmentation procedure. Also, in these sequences the viewpoint is fixed and the cars are moving in the same direction that increases the performance. Increasing the threshold value for matching costs can increase the recognition of similar parts but increases also the probability of false detection , i.e. decision of presence of OOI in an image without OOI.

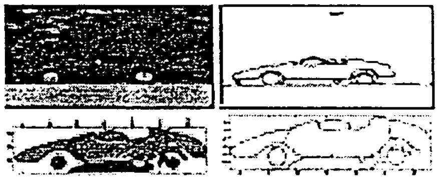


Figure 3. Top Left: Original image; Top Rigth: Color segmented image; Bottom Left: Scaled and curvature segmented object; Bottom Rigth: Resulting nodes of the object

In occlusion experiments, the effectiveness of the algorithm is tested using 10 images manually occluded. In order

[1] This publication includes some images from Corel Stock Photos which are protected by the copyright laws of the U.S., Canada and elsewhere. Used under license.

to obtain the performance in the presence of partial occlusion, the car objects in the images are partially removed. Figure 5 shows the results of the experiments and Figure 6 shows one example of occlusion experiments for still images and Figure7 one example from MPEG7 test sequence.

4 Conclusions and Future Work

In this paper, we have investigated a graph-based approach for object retrieval in digital image and video libraries. The results show that graph based methods that link low-level descriptions to high level semantic concepts can be efficiently used in text annotation of image and video databases. Object segmentation to its significant parts makes the approach more robust against segmentation errors and occlusion. In these cases, matching combination of reference nodes is implemented to reduce the probability of miss of correct matches. Since this approach is used for off-line annotation of images, the added time complexity has a secondary importance.

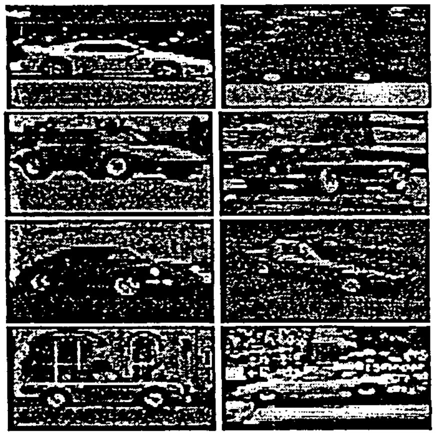


Figure 4. Some test Images

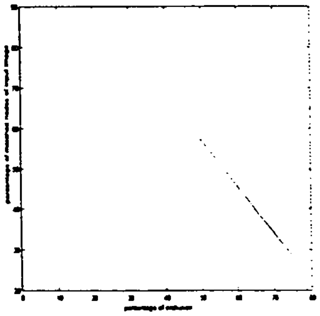


Figure 5. Occlusion experiments

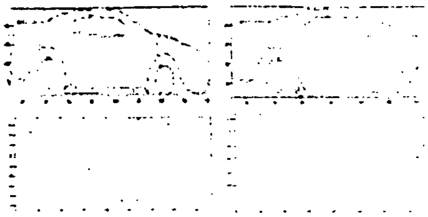


Figure 6. Resulting nodes of an occluded object (First image in Figure 4)

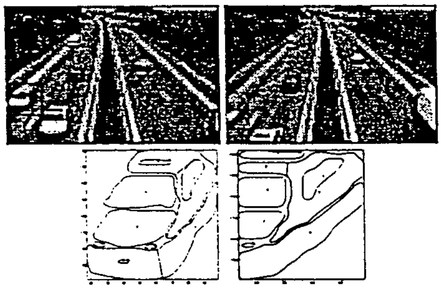


Figure 7. From MPEG-7 Video Content Set , Top Left: Sequence including non-occluded test car, Top Right: Sequence including occluded test car; Bottom Left: non-occluded car, Bottom Right: occluded car .

Acknowledgment

The first author would like to thank Dr. Aydin Alatan of New Jersey Institute of Technology for his invaluable discussions.

References

[1] B. Furht, S.W. Smoliar, and H.Zang, "Video and Image Processing in Multimedia System," Kluwer Academic Publishers, 1995.

[2] R.W. Picard and T.P. Minka, "Vision Texture for Annotation," MIT Multimedia Laboratory Perceptual Computing Section TR No.302, 1995.

[3] S. Scraloff and A. Pentland, "Modal Matching for Correspondence and Recognition," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 17, pp.545-561, 1995.

[4] H. Yu, W. Wolf, "A Visual Search System for Video and Image Databases," *Proc. IEEE Multimedia*, 1997.

[5] J. Zhang, H.Krim, and X. Zhang, "Invariant Object Recognition by Shape Space Analysis," *Proc. Int. Conf. on Image Proc.*, 1998.

[6] R. Jain, "Workshop Report: NSF Workshop on Visual Information Management Systems," *Proc. SPIE Conf. on Vis. Commun. and Image Proc.*, 1993.

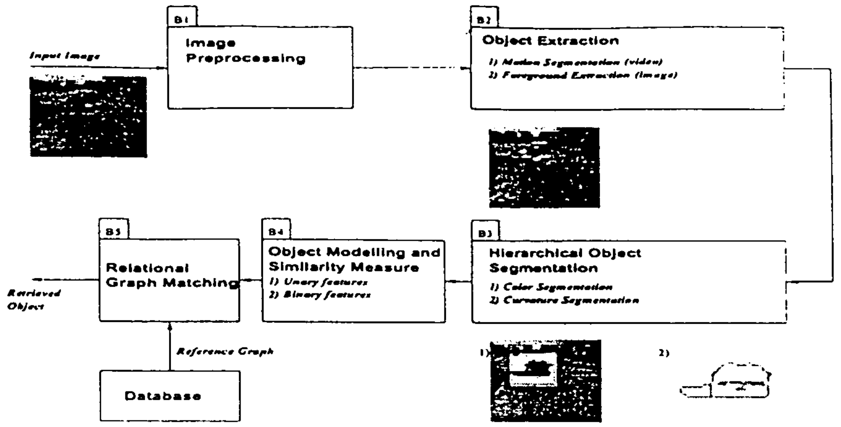

B1
Image Preprocessing
Input Image
B2
Object Extraction
1) Motion Segmentation (video)
2) Foreground Extraction (image)
B3
Hierarchical Object Segmentation
1) Color Segmentation
2) Curvature Segmentation
B4
Object Modelling and Similarity Measure
1) Unary features
2) Binary features
B5
Relational Graph Matching
Retrieved Object
Reference Graph
Database
1)
2)

Figure 8. Block diagram of the proposed retrieval system.

	# of test cars	# of correct detection	# of false detection	# of miss
MPEG 7	18	17	0	1
Still images A	23	19	0	4
Still images B	31	21	1	9

Table 1. System performance

[7] R. Jain, A. Pentland, and D. Petkovic, "NSF-ARPA Workshop on Visual Information Management Systems," Cambridge, MA, June 1995.

[8] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafine, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," *IEEE Computer*, 1995.

[9] J. Dowe, "Content-based Retrieval in Multimedia Imaging," *Proc. SPIE Conf. on Vis. Commun. and Image Proc.*, 1993.

[10] A. Pentland, R. Picard, and S. Scarloff, "Photobook: Content-based Manipulation of Image Databases," International Journal of Computer Vision, 1996.

[11] T.S. Huang, S. Mehrotra, and K. Ramchandran, "Multimedia Analysis and Retrieval System(MARS) Project," *Proc. of 33rd Annual Clinic on Library Application of Data Processing-Digital Image Access and Retrieval*, 1996.

[12] B. Gunsel and A.M. Tekalp, "Shape Similarity Matching for Query by Example," *Pattern Recognition*, vol. 31, No. 7, pp.931-944, July 1998.

[13] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge,"Comparing Images Using the Hausdorff Distance," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 15, pp.850-963, 1993.

[14] F.S. Cohen, Z. Huang, and Z. Yang," Invariant Matching and Identification of Curves Using B-splines Curve Representation," *IEEE Trans. on Image Processing*, vol. 4, pp.1-10, 1995.

[15] E. Persoon and K.S. Fu, "Shape Discrimination Using Fourier Descriptors," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 8, pp.388-397, 1986.

[16] J.L. Mundy and A. Zisserman, "Geometric Invariance in Computer Vision," MIT Press, 1992.

[17] M.P. Dubuisson, A.K. Jain, and W.C. Taylor, "Segmentation and Matching of Vehicles in Road Images," *Transportation Research Report*, No.1412, pp.57-63.

[18] D.H. Ballard and C.M. Brown, "Computer Vision," Prentice-Hall, Englewood Cliffs, NJ, 1982.

[19] T. Caelli and W.F. Bischof, "Machine Learning and Image Interpretation," Plenum Press, New York, NY, 1997.

[20] J. Shi and C. Tomasi, "Good Features to Track," *CVPR*, 1994.

[21] K. Harris, S.N. Efstratiadis, N. Maglaveras, and A.K. Katsaggelos, "Hybrid Image Segmentation Using Water Sheds and Fast Region Merging," *IEEE Trans. on Image Processing*, vol. 7, pp.1684-1699, 1998.

[22] F. Mokhtarian and A. Mackworth, "Scale Based Description and Recognition of Planar Curves and 2D Shapes," *IEEE PAMI*, vol. 8(1), pp.34-43, 1986.

[23] R.M. Haralick and L.G. Shapiro, "Computer and Robot Vision," Addison Wesley Publishing Co., 1993.

[24] J. B. Burns, R. S. Weiss and E. M. Riseman, "View Variation of Point-Set and Line-Segment Features," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, No 1, pp. 51-68, 1993